

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA LA
EMULACIÓN DE FUENTES DE ALIMENTACIÓN EN
LAZO CERRADO**

Javier Casatorres Agüero

Tutor: Ángel de Castro Martín

Julio 2015

DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA LA EMULACIÓN DE FUENTES DE ALIMENTACIÓN EN LAZO CERRADO

AUTOR: Javier Casatorres Agüero

TUTOR: Ángel de Castro Martín



Human Computer Technology Laboratory

Dpto. TEC

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2015

AGRADECIMIENTOS

A Ángel, mi tutor, por las oportunidades que me ha ofrecido.

A mis amigos, por ser mis amigos.

A todos aquellos que saben qué significa de verdad la palabra *placa*.

Y a mis padres y a mi hermano, por tantas razones que no entrarían en mil páginas.

Gracias.

RESUMEN

El desarrollo del presente Trabajo de Fin de Grado ha perseguido diversos objetivos relacionados con la electrónica de control, y más concretamente, con el uso de la técnica *Hardware In The Loop* como mecanismo para solucionar el problema que suponen las simulaciones mixtas analógico-digitales de los sistemas de control de fuentes conmutadas. La técnica de *Hardware In The Loop* permite emular el sistema completo en hardware pudiendo llegar a funcionar en tiempo real o a velocidades muy superiores a las que permiten las simulaciones mixtas, evitando los problemas derivados de un malfuncionamiento en el control de una planta real.

En el contexto que establece esta técnica, se ha realizado el desarrollo de un modelo HIL para un convertidor conmutado de alta frecuencia de conmutación. Esta elevada frecuencia ha forzado al uso de señales de coma fija y de FPGA como soporte para la síntesis del modelo digital.

En primer lugar se ha diseñado en lenguaje VHDL y sintetizado sobre FPGA el modelo digital de un convertidor DC-DC reductor cuadrifase de tipo Buck en coma fija con una frecuencia de conmutación de 200 kHz. La discretización de las ecuaciones diferenciales que rigen el funcionamiento del circuito, la optimización de la resolución de las señales implicadas evitando la pérdida de precisión, y la aplicación de mecanismos conducentes a que el modelo pudiese ser sintetizado en el modelo de FPGA empleado han sido claves en esta tarea.

Paralelamente y en estrecha relación con lo anterior, se ha desarrollado y construido el módulo de conversión digital analógico que permite completar el modelo HIL de la planta transformando las señales digitales del modelo digitalizado en señales analógicas como las que generaría la planta real. El circuito se ha diseñado permitiendo que pueda ser usado con cualquier modelo digital de planta cuyas señales de salida tengan una frecuencia máxima de 100 MHz. Una amplia batería de pruebas ha permitido seleccionar los componentes y configuraciones que mejor se ajustan a esta aplicación con el objeto de maximizar su rendimiento, cuestión que ha culminado con la integración del sistema completo y su comparación con un convertidor Buck real.

Como objetivo adicional, una colaboración con la empresa SpControl Technologies se ha concretado en diseñar la placa de conversión mencionada de forma que sea compatible con la placa de control *SpCard* que ofrece la empresa. El buen funcionamiento del circuito diseñado en las pruebas a las que se someta podría llevar a su integración como un módulo adicional de la *SpCard*, con su correspondiente explotación comercial.

PALABRAS CLAVE

Hardware In the Loop, Control digital, Fuentes de alimentación conmutadas, Convertidor Buck, Conversión digital-analógica.

ABSTRACT

The development of this Final Degree Thesis has been done keeping several control-electronics related objectives in mind. More specifically, it is focused on the *Hardware In the Loop* technique usage, as a mean to solve the issues that appear when doing mixed signals simulation of switching-mode power supplies. The *Hardware in the Loop* technique allows a full hardware emulation reaching high speed or even real-time operations while avoiding the problems that may appear with a real-plant-control malfunction.

In this context, it has been realized the whole design of a HIL model for a switch-mode converter with a high switching frequency. This high operating frequency has forced the use of FPGA and fixed point signals.

Firstly a four-phase DC-DC Buck converter digital model with a 200 kHz switching frequency has been designed using VHDL and then implemented over FPGA. Discretizing the differential equations that model the behavior of the circuit, optimizing the number of bits of every fixed-point signal and applying mechanisms to make the synthesis and implementation possible have been the main tasks.

At the same time and closely related to the formerly mentioned, it has been developed the digital to analog conversion module, which makes the HIL model complete by transforming the digital signals from the digital module into analog signals similar to those generated by the real plant. The circuit has been designed so that it can be used with any digital model as long as its output signals have a maximum frequency of 100 MHz. A broad range of tests has led to selecting the components and structures that work best in order to maximize performance, and has finalized with the integration of the full HIL model, which has then been compared with a real Buck converter.

As an additional objective, a collaboration with SpControl Technologies has meant that the conversion circuit has been designed to make it compatible with the firm's control board *SpCard*. A good performance of the circuit when tested might mean its integration as an additional module of the *SpCard* and its subsequent commercial use.

KEYWORDS

Hardware In the Loop, Digital control, Switching-mode power supplies, Buck converter, Digital to analog conversion.

Índice general

CAPÍTULO 1 : Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura de la memoria	2
CAPÍTULO 2 : Estado del Arte	3
CAPÍTULO 3 : Desarrollo de un módulo de conversión	9
3.1. Estructura del diseño	9
3.2. Selección de componentes	10
3.2.1. Selección de DACs	10
3.2.2. Selección de amplificadores operacionales	14
3.2.3. Otros componentes.....	16
3.3. Diseño y evaluación de placas separadas	17
3.3.1. Placa de conversión.....	18
3.3.2. Placa de amplificación	21
3.3.3. Pruebas.....	24
3.3.4. Resultados	26
3.4. Diseño y evaluación de placas conjuntas	30
3.4.1. Compatibilidad con SpCard	31
3.4.2. Uniformización de pistas	33
3.4.3. Pruebas y resultados de placas conjuntas	34
CAPÍTULO 4 : Diseño del modelo digitalizado de un convertidor Buck	37
4.1. Convertidor Buck, ecuaciones y discretización	37
4.2. Modelo REAL con reloj de 50 MHz	39
4.2.1. Diseño unifase	39
4.2.2. Diseño cuadrifase	41
4.3. Simulaciones del circuito analógico	42
4.4. Modelo cuadrifase en coma fija a 50 MHz.....	43
4.4.1. Biblioteca sfixed y primer modelo en coma fija	43
4.4.2. Estudio de resolución y modelo optimizado	44
4.5. Modificaciones para síntesis	46
4.5.1. Soluciones al problema de cumplimiento de tiempos.....	47
CAPÍTULO 5 : Integración final	51
CAPÍTULO 6 : Conclusiones y trabajo futuro	55
6.1. Conclusiones.....	55

6.2. Trabajo futuro	56
CAPÍTULO 7 : Bibliografía	57
ANEXO A Pinout y huella de los componentes	i
Convertor digital-analógico AD9752ARZ	i
Convertor digital-analógico AD9754ARZ	i
Convertor digital-analógico DAC904U	i
Convertor digital-analógico THS5651AIDW	ii
Amplificador operacional CLC1005	ii
Amplificador operacional ADA4857	ii
ANEXO B Diseños en Altium	iii
Placa de conversión común.....	iii
Placa de conversión THS5651AIDW	v
Placa de amplificación inversora.....	vii
Placa de amplificación no inversora.....	viii
Placas conjuntas.....	ix
ANEXO C Generadores de señales de prueba	xiii
Generador de escalones.....	xiii
Generador de señales	xiii
Asignación de pines.....	xv
ANEXO D Pruebas con señales	xvii
Seno a 50 kHz	xvii
Seno a 1 MHz.....	xix
Triangular a 50 kHz.....	xxi
Triangular a 1 MHz	xxiii
ANEXO E Código VHDL de modelos digitalizados	xxv
Modelo REAL unifase	xxv
Testbench del modelo REAL unifase	xxvii
Modelo cuadrifase en coma fija con segmentación	xxix
Generador de PWMs desfasados con reloj de 50 MHz.....	xxxiii
Modelo cuadrifase en coma fija a 20 MHz.....	xxxv
Modelo de alimentación y carga variable	xxxix
ANEXO F Simulaciones del transitorio de V_{out}	xli
Simulación analógica del modelo unifase	xli
Modelo REAL unifase a 50 MHz	xli
Simulación analógica del modelo cuadrifase	xli
Modelo REAL cuadrifase a 50 MHz	xlii

Modelo cuadrifase a 50 MHz en coma fija.....	xlii
Modelo cuadrifase a 50 MHz en coma fija con segmentación	xlii
Modelo cuadrifase a 20 MHz en coma fija.....	xliii
ANEXO G Comparativa modelo HIL – realidad.....	xlv
Corriente de salida de 1 A	xlv
Corriente de salida de 2,5 A	xlvi
Corriente de salida de 5 A	xlvi
Corriente de salida de 8 A	xlvi
ANEXO H Coste del módulo de conversión completo.....	xlix

Índice de figuras

Figura 1: Esquema de un convertidor Buck unifase	2
Figura 2: Estructura de un sistema en lazo cerrado	3
Figura 3: Esquema general de la técnica Hardware In the Loop.....	4
Figura 4: HIL en procesos industriales [HILSim]	5
Figura 5: Modelo completo de una fuente de alimentación.	6
Figura 6: Estructura del circuito	10
Figura 7: Configuración de salida single-ended típica.....	19
Figura 8: Esquemático de la placa de conversión para el AD9754ARZ	20
Figura 9: Esquemático de la placa de conversión para el AD9754ARZ	20
Figura 10: Conversor de corriente a tensión.....	21
Figura 11: Inversor simple con operacional	22
Figura 12: Etapa de amplificación en configuración inversora	22
Figura 13: Circuito conversor corriente a tensión, con amplificador no inversor	23
Figura 14: Etapa de amplificación en configuración no inversora	24
Figura 15: Generadores de señales	25
Figura 16: Ruido en configuración inversora	27
Figura 17: Ruido en una señal continua.	28
Figura 18: Seno a 50 kHz	28
Figura 19: Resultado de sustituir el ADA4857 por el CLC1005	28
Figura 20: Espectro de una senoide de 50 kHz.....	30
Figura 21: Señal triangular de 1 MHz en el tiempo.....	30
Figura 22: Pinout de conectores de la SpCard	31
Figura 23: Pinout del conector	31
Figura 25: Modificaciones en salidas digitales para permitir compatibilidad.....	32
Figura 26: Configuración de salida	33
Figura 27: Uniformización de longitud de las pistas	34
Figura 28: Seno de 1 MHz	35
Figura 29: Montaje del sistema.....	36
Figura 30: Convertidor Buck unifase	38
Figura 31: Buck multifase (3 fases).....	41
Figura 32: Interleaving en un modelo de dos fases.	42
Figura 33: Circuito del Buck en Simulink	43
Figura 34: Corrientes del modelo cuadrifase.	43
Figura 35: Esquema simplificado del hardware interviniente en el modelo HIL	46
Figura 36: Sistema completo apto para ser sintetizado.....	47
Figura 37: Regimen transitorio de la tensión de salida, con los instantes cuyos valores se comparan	49
Figura 38: Transitorio de la tensión de salida con una corriente de salida de 8 A.	52
Figura 39: Comparativa de transitorios para corriente de salida de 8 A.	53
Figura 40: Pinout y huella del AD9752ARZ.....	i
Figura 41: Pinout y huella del AD9754ARZ.....	i
Figura 42: Pinout y huella del AD9752ARZ.....	i
Figura 43: Pinout y huella del THS561AIDW	ii
Figura 44: Pinout y huella del CLC1005	ii
Figura 45: Pinout y huella del ADA4857.....	ii

Figura 46: Esquemático de la placa de conversión común	iii
Figura 47: Layout de la placa de conversión común	iv
Figura 48: Esquemático de la placa de conversión del DAC THS5651AIDW	v
Figura 49: Layout de la placa de conversión del DAC THS5651AIDW	vi
Figura 50: Esquemático de la etapa de amplificación en configuración no inversora.....	vii
Figura 51: Layout de la etapa de amplificación en configuración no inversora	vii
Figura 52: Esquemático de la placa de amplificación en configuración no inversora	viii
Figura 53: Layout de la placa de amplificación en configuración no inversora	viii
Figura 54: Esquemático de las placas conjuntas	x
Figura 55: Layout de la placa con junta con uniformización de longitud de pistas	xi
Figura 56: Layout de la placa conjunta sin uniformización de pistas.....	xii
Figura 57: Seno en el tiempo a 50 kHz.....	xvii
Figura 58: Espectro del seno a 50 kHz en banda ancha	xviii
Figura 59: Espectro del seno a 50 kHz en banda estrecha	xviii
Figura 60: Seno en el tiempo a 1 MHz	xix
Figura 61: Espectro del seno a 1 MHz en banda ancha	xx
Figura 62: Espectro del seno a 1 MHz en banda estrecha	xx
Figura 63: Triangular en el tiempo a 50 kHz.....	xxi
Figura 64: Espectro de señal triangular de 50 kHz en banda ancha	xxii
Figura 65: Espectro de señal triangular de 50 kHz en banda estrecha	xxii
Figura 66: Triangular en el tiempo a 50 kHz.....	xxiii
Figura 67: Espectro de la señal triangular de 1 MHz en banda ancha	xxiv
Figura 68: Espectro de la señal triangular de 1 MHz en banda estrecha.....	xxiv
Figura 69: Transitorio de tensión de salida de la simulación analógica unifase	xli
Figura 70: Transitorio de tensión de salida del modelo REAL unifase	xli
Figura 71: Transitorio de tensión de salida de la simulación analógica cuadrifase	xli
Figura 72: Transitorio de tensión de salida del modelo REAL cuadrifase	xlii
Figura 73: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 50 MHz	xlii
Figura 74: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 50 MHz con segmentación	xlii
Figura 75: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 20 MHz	xliii
Figura 76: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 1 A.....	xlvi
Figura 77: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 2,5 A	xlvi
Figura 78: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 5 A	xlvi
Figura 79: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 8 A	xlvi

Índice de tablas

Tabla 1: Tabla de DACs candidatos (I)	13
Tabla 2: Tabla de DACs candidatos (II)	13
Tabla 3: Características DACs seleccionados (I)	14
Tabla 4: Características de los DACs seleccionados (II)	14
Tabla 5: Características de los amplificadores operacionales	16
Tabla 6: Comparativa de modelos digitalizados (I)	50
Tabla 7: Comparativa de modelos digitalizados (II)	50
Tabla 8: Comparativa de modelos digitalizados (III)	50
Tabla 9: Coste del módulo de conversión	xlix

Índice de código

Código 1: Selección de modo de funcionamiento	40
Código 2: Implementación de ecuaciones en diferencias	40
Código 3: Modificaciones en las ecuaciones en diferencias para el modelo cuadrifase	42
Código 4: Ejemplos de definición de tipo sfixed	44
Código 5: Uso de to_slv, to_sfixed y resize para asignar señales a puertos y viceversa	44
Código 6: Especificación de la condición de tiempo que debe cumplir el circuito	46
Código 7: Implementación del módulo anti-rebote	51
Código 8: Generador de señales escalón	xiii
Código 9: Generador de señales	xiv
Código 10: Asignación de pines	xv
Código 11: Modelo digitalizado en formato REAL del Buck unifase	xxvi
Código 12: Testbench del modelo REAL unifase	xxviii
Código 13: Modelo en coma fija con segmentación a 50 MHz	xxxiii
Código 14: Generador de PWMs con reloj a 50 MHz	xxxv
Código 15: Modelo cuadrifase en coma fija a 20 MHz	xxxix
Código 16: Modelo de alimentación y carga variable para el Buck	xxxix

CAPÍTULO 1: Introducción

1.1. Motivación

El enorme desarrollo que ha tenido el control digital de sistemas ha propiciado la aparición de una técnica de prueba de los reguladores implicados en el proceso: la llamada técnica de *Hardware In the Loop* o simplemente HIL.

Esencialmente, la idea se basa en sustituir el elemento a regular o *planta*, que puede ser cualquier tipo de sistema (mecánico, hidráulico, electrónico...), por un *modelo* de su funcionamiento (llamado modelo HIL) que se ejecuta sobre hardware en tiempo real, de forma que para el regulador sea indistinguible si está controlando una planta real o dicho modelo. En consecuencia, el funcionamiento del regulador puede evaluarse conectándolo directamente al modelo HIL, con la ventaja de evitar los problemas y daños materiales o personales que podría conllevar un malfuncionamiento en el control de una planta real.

En el caso de los sistemas electrónicos, esta técnica cobra especial relevancia cuando se aplica al modelado digital de fuentes de alimentación. Un modelo HIL de una planta de estas características está constituido por un modelo digital sintetizable, generalmente escrito en algún lenguaje de descripción hardware; y un módulo capaz de transformar las señales digitales de ese modelo en señales analógicas como las que produciría la planta real. Mientras que los simuladores de sistemas mixtos como el formado por un regulador *digital* y su planta *analógica* son caros, complejos y sobre todo muy lentos, cuando se desarrolla un modelo HIL de una planta analógica, dicho modelo puede emularse en hardware, lo que permite comprobar el funcionamiento del regulador funcionando en tiempo real o a velocidades muy superiores a las ofrecidas por simulaciones.

Las señales intervinientes en el modelo digital de la planta suelen tener valores no enteros, por lo que se hace necesaria la utilización de un sistema de representación de números decimales. Existen dos opciones posibles: representación en coma flotante o en coma fija. La primera es mucho más compleja, lenta y requiere mucho más hardware para funcionar. A cambio, la representación en coma fija supone una mayor dificultad para el diseño, pues debe realizarse un estudio de resolución que permita determinar cuántos bits se dedican a cada señal. Frente a los más grandes y complejos sistemas en que las señales suelen ser lentas en formato de coma flotante, existen ciertas aplicaciones, como la regulación de fuentes conmutadas de elevada frecuencia de conmutación, en las que el uso de la coma fija no puede evitarse por la alta velocidad de los procesos implicados. Este campo de diseño, que actualmente está casi totalmente restringido a la investigación, es uno de los pilares del presente Trabajo de Fin de Grado.

1.2. Objetivos

El propósito del trabajo que se presenta es desarrollar el modelo HIL completo de una determinada planta analógica. Para ello, en primer lugar se desarrolla el circuito completo de conversión digital-analógico buscando que sea totalmente independiente del modelo digitalizado de planta que deba complementar, analizando las necesidades y seleccionando los componentes y configuraciones que mejores prestaciones ofrecen para ese propósito.

Por otro lado y para poder completar el sistema, se crea y sintetiza sobre FPGA el modelo digitalizado en coma fija de una planta analógica en VHDL, concretamente de un convertidor DC-DC reductor multifase de tipo “Buck”, con una frecuencia de conmutación de 200 kHz, cuyo esquema simplificado unifase se muestra en la Figura 1. Debido a que en cada ciclo de conmutación se necesitan del orden de 100 ciclos de cálculo para permitir que el modelo sea fiel a la realidad, aparecen señales con frecuencias del orden de megahercios, lo que obliga al uso de señales en formato de coma fija.

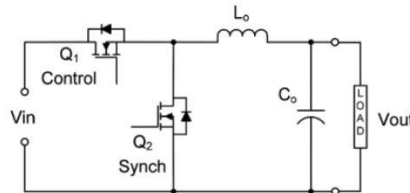


Figura 1: Esquema de un convertidor Buck unifase

Adicionalmente, cabe destacar que el módulo de conversión creado se ha diseñado de forma que sea compatible con tecnología comercial procedente de la empresa *SpControl Technologies* [SpControl], que colabora con el grupo de investigación *HCTLab* [HCTLab], en que el presente trabajo se ha realizado. Más concretamente, el módulo se ha diseñado para que pueda ser conectado directamente a la placa de prototipado *SpCard*, y posteriormente integrado en ella si las prestaciones que ofrece son adecuadas.

1.3. Estructura de la memoria

A lo largo del CAPÍTULO 2 se realiza una explicación más detallada del funcionamiento de HIL, dándose ciertos ejemplos de su integración en el mercado. Se ofrece después una breve descripción sobre las fuentes de alimentación conmutadas y los formatos de representación de valores decimales, ámbitos directamente relacionados con el trabajo que se presenta.

En el CAPÍTULO 3, se describe en detalle el proceso de selección de componentes para la creación de la placa de conversión. El diseño de la estructura y del esquemático del circuito se desarrolla después, indicando las decisiones de diseño tomadas. Finalmente, se muestran los resultados obtenidos tras construir las placas de circuito impreso, soldar los componentes y someter los circuitos resultantes a un proceso de pruebas con generadores de señales.

El CAPÍTULO 4 se centra en el desarrollo del modelo digitalizado de un convertidor DC-DC reductor tipo Buck multifase en coma fija. Con este objetivo, se describe primero el desarrollo de un modelo unifase con señales en un formato no sintetizable, las comparaciones de resultados con simulaciones del circuito analógico y la utilidad del uso de varias fases. El paso a coma fija y los problemas surgidos en la síntesis del modelo sobre FPGA cierran el capítulo.

A continuación, y una vez se tienen los dos bloques funcionando correctamente, se integran para comprobar el buen funcionamiento del sistema completo, cuestión descrita en el CAPÍTULO 5, en el que se incluye además una comparación del funcionamiento del sistema completo con respecto al de un convertidor real.

Finalmente se presentan las conclusiones y las posibles líneas de trabajo futuro, lo que constituye el CAPÍTULO 6.

CAPÍTULO 2: Estado del Arte

Entre las aplicaciones de la electrónica, el control automático ocupa un puesto muy destacado. Esta disciplina, que no es exclusiva de la electrónica, se centra en el desarrollo de sistemas capaces de gobernar su propia actuación independientemente de las características que los rodeen, llegando a corregir de forma autónoma los errores que puedan aparecer.

Pese a que la planta de un sistema de control es capaz de realizar por sí sola la función del sistema, funcionando sin regulador en lo que se conoce como *lazo abierto*, generalmente su funcionamiento aislado lleva a determinados errores, tanto en el valor final (como ocurre si se pide a un sistema que se mueva a una determinada posición, y el sistema termina por no llegar o pasarse), como en el tiempo que tarda en llegar a ese valor final, período conocido como *régimen transitorio* (por ejemplo, es lo que ocurre si se pide al sistema que se mueva a una determinada posición sin pasarse en ningún momento, y el sistema oscila: primero se pasa de la posición pedida, después retrocede demasiado, y vuelve a avanzar demasiado en un proceso que se repite de forma cada vez más atenuada hasta que se llega a la posición pedida).

Para modificar el funcionamiento aislado de la planta, se conecta a la misma un regulador formando un sistema realimentado como el mostrado en la Figura 2. Se dice entonces que el sistema funciona en *lazo cerrado*. Si el regulador se diseña correctamente, será capaz de evitar los problemas comentados anteriormente modificando la señal que le llega a la planta en función de la salida del sistema.

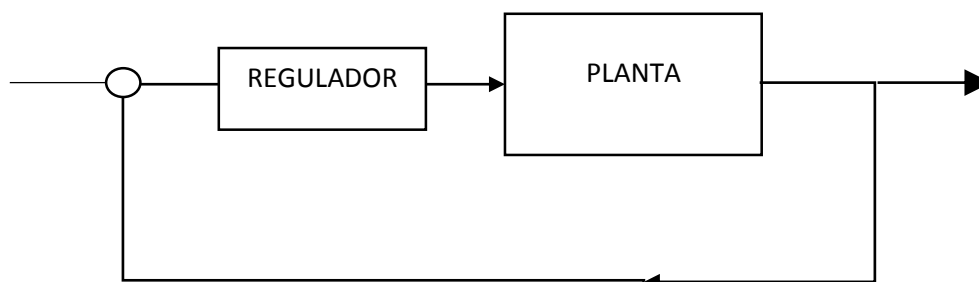


Figura 2: Estructura de un sistema en lazo cerrado

La mayor parte de las plantas que existen son sistemas mecánicos o químicos de procesos industriales. Sin embargo, existen ciertas plantas formadas por circuitos eléctricos o electrónicos, y generalmente son de carácter analógico. Un ejemplo es el encarnado por las fuentes de alimentación, y más concretamente, por los convertidores conmutados, en los que se centra este trabajo.

De la misma manera, los reguladores pueden estar formados por sistemas de diversa índole (mecánicos, hidráulicos...). Sin embargo, en el control de circuitos eléctricos o electrónicos lo habitual es encontrarse reguladores constituidos a su vez por otros circuitos eléctricos o electrónicos.

Tradicionalmente, las señales que intervenían en el funcionamiento de un regulador de estas características eran analógicas. Sin embargo, con el enorme desarrollo de la electrónica digital de los últimos años, este hecho ha cambiado: muchas aplicaciones incorporan ahora reguladores digitales. Esto ofrece multitud de ventajas, como la posibilidad de funcionar más rápido y con menos errores, y una mayor independencia de las condiciones externas, o incluso

un menor coste en determinados casos. Un ejemplo muy ilustrativo de las ventajas que ofrece la electrónica digital se refiere a las modificaciones que sufre el valor de los elementos pasivos con los cambios de temperatura, que pueden llevar a que un sistema analógico no funcione correctamente, y que no afectan a un sistema digital, sin elementos pasivos críticos para su funcionamiento.

A cambio, existen ciertos problemas derivados del paso al mundo digital. Además de la necesidad de introducir conversores analógico-digital y digital-analógico en el sistema para permitir la correcta comunicación entre planta y regulador, un problema es la simulación del sistema funcionando en *lazo cerrado*. Como es lógico, los reguladores no deben ser implementados en un sistema de control real hasta que hayan sido probados debidamente, máxime en aplicaciones de potencia en la que errores de control pueden provocar malfuncionamientos que destruyan el sistema y dañen a las personas cercanas.

Las simulaciones son, consecuentemente, imprescindibles, y requieren un simulador mixto, pues la planta es analógica en la gran mayoría de los casos, frente a un regulador digital. Como ya se ha dicho, estas simulaciones son muy lentas, de forma que pocos segundos simulados pueden llevar muchas horas de tiempo real [Sanchez]. Alternativamente, hay propuestas basadas en simular las partes digitales y analógicas por separado [Sanchez] [Zumel]. El problema radica en que hay que desarrollar una interfaz entre los resultados de ambas partes usando algún lenguaje de programación, lo que no es tarea fácil.

Ya se ha dicho que la técnica de *Hardware In The Loop*, cuyo funcionamiento se muestra en la Figura 3, puede solucionar el problema a través de la emulación hardware. Una ventaja adicional que ofrece HIL es que el sistema completo en lazo cerrado puede simularse a través de un simulador digital, lo que disminuye enormemente el tiempo necesario. Sin embargo, esta ventaja suele quedar eclipsada por el aumento de velocidad que ofrece la emulación del sistema directamente en hardware, que es mucho mayor (llegando, según se ha comentado, a funcionar en tiempo real) que el que ofrece una simulación digital del lazo completo.

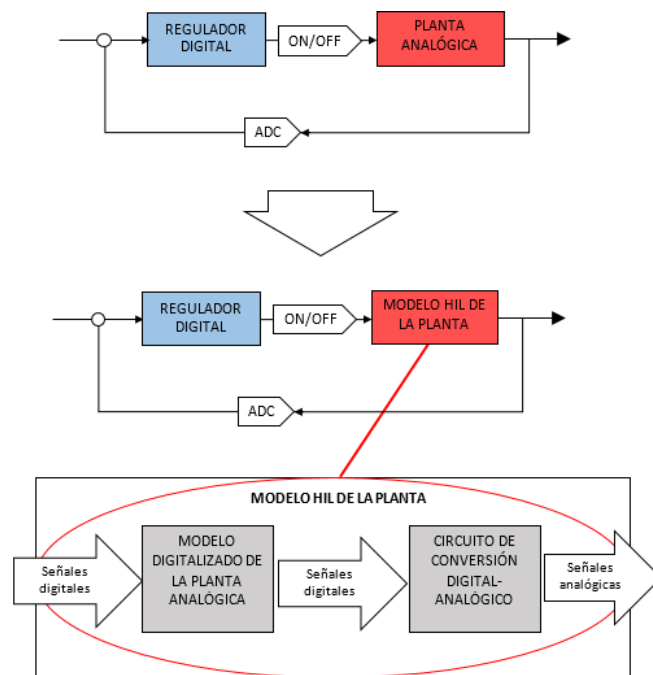


Figura 3: Esquema general de la técnica Hardware In the Loop

Según se aprecia en la Figura 3, en muchas ocasiones el regulador envía a la planta señales de carácter digital (ON/OFF o señales de modulación de ancho de pulso *PWM*) encargadas de permitir que conduzcan o no determinados elementos de control (habitualmente transistores de efecto campo de semiconductor de óxido de metal o *MOSFETs* por sus siglas en inglés). Sin embargo, hay ciertos sistemas en que esto no ocurre así: la señal que debe recibir la planta para su control es de carácter analógico, y por ello suele requerirse un conversor digital-analógico (*DAC*) entre la salida del regulador y la planta. Si así fuera, en el modelo HIL de la planta sería necesario un conversor analógico-digital que permitiera que las señales que entrasen al modelo digitalizado de la planta fuesen digitales. El modelo desarrollado en este trabajo es del primer tipo: el regulador controla el funcionamiento de la planta a través de señales *PWM* que permiten la conducción/no conducción de *MOSFETs*, así que se sigue el esquema exacto de la Figura 3.

Las ventajas de HIL mencionadas antes explican el enorme desarrollo que ha tenido la técnica, que ha llegado a ocupar un lugar destacado en el ámbito del control de procesos. Existen numerosos ejemplos de empresas que han encontrado un nicho de mercado en el desarrollo de tecnología relacionada con HIL y emulaciones de sistemas en tiempo real, como *OPAL-RT technologies* [Opal] o *dSPACE* [dSPACE]. Pese a todo, se trata de una tecnología muy joven (en torno a 15-20 años), inicialmente enfocada a sistemas muy complejos, como los que existen en industria aeroespacial, donde apareció originalmente como respuesta al aumento de los costes derivados del desarrollo de prototipos para pruebas cada vez más complejos, que en muchos casos eran destruidos.

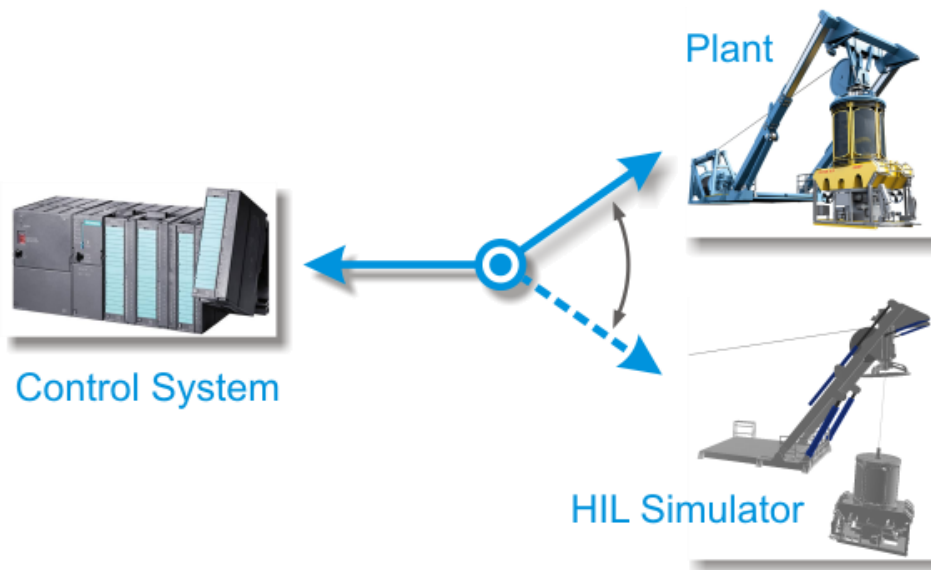


Figura 4: HIL en procesos industriales [HILSim]

En este tipo de sistemas, las señales que intervienen son lentas, y los costes de la tecnología asociada son muy elevados. Un ejemplo revelador de esto último es el precio de un producto para realizar modelos HIL en alto nivel: el aparato más la licencia de software anual cuesta más de 15.000 €. Por otro lado, las frecuencias de cálculo de los nuevos valores no suelen superar frecuencias del orden del megahercio.

Únicamente en los últimos años ha comenzado la expansión de la tecnología a sistemas en los que la velocidad de las señales es alta y por tanto se requieren mayores frecuencias de

cálculo. La aplicación de la técnica HIL a la electrónica de potencia ha supuesto un nuevo trampolín para el desarrollo de la primera, y abre numerosos mercados y vías de expansión, que ya están explotando empresas punteras como *Typhoon-HIL* [Typhoon].

Se abre así un nuevo camino en el que la complejidad de modelos cambia por un gran aumento en la velocidad de las señales implicadas, lo que obliga a cambiar los sistemas donde se implementan los modelos digitales. Frente a potentes procesadores capaces de modelar los sistemas más complejos en software, aparece la emulación hardware sobre FPGA o elementos similares, capaces de trabajar a frecuencias enormemente superiores debido a su inherente capacidad de trabajo paralelo.

Las fuentes de alimentación con reguladores de tensión conmutados de alta frecuencia de conmutación, omnipresentes en todo tipo de aparato electrónico en la actualidad, constituyen un perfecto nicho para el desarrollo de esta tecnología. La frecuencia de conmutación es uno de los parámetros clave en el rizado de la tensión de salida de este tipo de fuentes, y si es elevada fuerza a que la frecuencia de cálculo de un modelo HIL lo sea mucho más, debiendo por tanto implementarse necesariamente sobre FPGA, y en muchos casos usando señales en formato de coma fija. Éste es actualmente un ámbito restringido casi exclusivamente a la investigación.

El funcionamiento de una fuente de alimentación se basa en la concatenación de varias etapas, según se muestra en la Figura 5. Su propósito es transformar la tensión alterna de la red eléctrica en una tensión continua de un valor determinado para posibilitar su aprovechamiento por otros circuitos electrónicos.

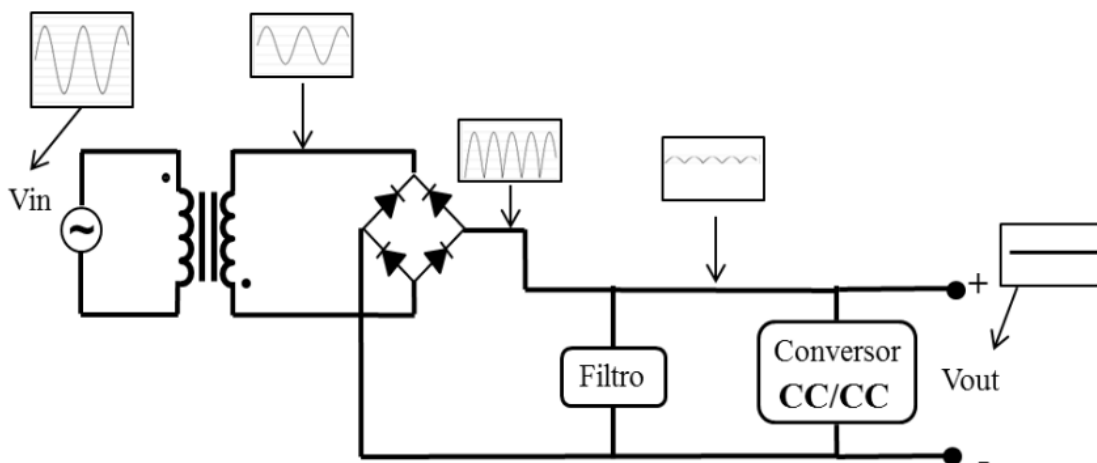


Figura 5: Modelo completo de una fuente de alimentación.

Con este objetivo, el primer elemento del sistema es un transformador, que reduce la amplitud de la tensión de entrada a un valor mucho menor. Aunque de menor amplitud, la señal de salida del transformador sigue siendo sinusoidal, siendo necesario rectificarla. Para ello se utiliza un puente de diodos que funciona como rectificador de onda completa, consiguiendo que la tensión tenga siempre valores positivos. A consecuencia de la rectificación, la señal resultante ya no tiene únicamente componente alterna, sino que su valor medio deja de ser nulo, o lo que es lo mismo, aparece una componente continua. Queda eliminar la mayor parte posible de componente alterna, para lo cual se utiliza un filtro paso bajo (generalmente, un condensador de elevado valor en paralelo con la salida del puente de diodos).

Con el proceso completo se consigue una señal aproximadamente continua, pero con un rizado dependiente del valor del condensador del filtro de eliminación de componente alterna y su resistencia serie equivalente (ESR), rizado que en muchos casos es inadmisibles. Paralelamente, el valor de la señal de salida suele ser superior al deseado. Es aquí donde se utiliza un regulador de tensión, capaz de ofrecer una tensión casi constante a su salida a partir de una señal variable en su entrada.

Existen dos grandes grupos de reguladores de tensión con características muy diferentes: lineales y conmutados. El grupo de interés para este trabajo está formado por los convertidores conmutados, que ya han sido presentados anteriormente. Se trata de reguladores de tensión mucho más complejos que los lineales, requiriendo de un circuito de control adicional, pero son a cambio mucho más eficientes, pues no disipan el exceso de tensión en forma de calor. De esta forma, además de evitar el calentamiento y requerir por tanto de grandes disipadores, permiten un gran ahorro energético.

El funcionamiento de los convertidores conmutados se basa en la conmutación entre los estados de conducción y no conducción de ciertos MOSFETs, siendo la *frecuencia de conmutación* la velocidad con que ambos estados se alternan. Por otro lado, la relación entre el estado de conducción y el de no conducción determina el valor de la señal de salida, y suele ser controlada por un regulador externo.

Para poder comprobar el buen funcionamiento de dicho regulador, una posibilidad es realizar, según se ha comentado, un modelo HIL completo del convertidor. Así puede probarse el sistema en lazo cerrado y en tiempo real sin peligro de daños materiales o personales en caso de un malfuncionamiento.

En aplicaciones en que el rizado debe ser reducido, se hace necesario que la frecuencia de conmutación sea alta (puede considerarse que la conmutación introduce un ruido de frecuencia igual a la de la frecuencia de conmutación, y que cuanto mayor sea la frecuencia de ésta, más atenuado quedará el ruido gracias a un filtrado paso bajo posterior). Cuando se realiza un modelo HIL del convertidor, se hacen necesarias señales de alta velocidad, y conseguir que haya muchos ciclos de cálculo por cada ciclo completo de simulación, de forma que el modelo digitalizado sea lo más cercano posible a la realidad. Esto significa, según se ha adelantado, que será necesario implementarlo sobre FPGA y usando señales que permitan alta velocidad.

Existen dos formas posibles de representar señales con valores decimales. Los estándares de coma fija permiten establecer una determinada cantidad de bits destinados a representar la parte entera de la señal, y otra para representar la parte fraccionaria, todo mediante potencias de dos, positivas en el caso de la parte entera, y negativas en el caso de la fraccionaria. De esta forma, se aproxima el valor real de la señal al de la suma de potencias de dos, positivas y negativas, más cercana a éste dentro de los límites que establece el número de bits. Si además el estándar tiene en cuenta el signo, generalmente el número estará en complemento a 2, reservándose el bit más significativo para representar el signo.

La alternativa a esto es el uso de coma flotante. Existen numerosos estándares que definen formas para computar coma flotante, pero uno de los más extendidos es el IEEE 754, que define el uso de un número fijo de bits para la representación de un número no entero. De esos bits, una fracción se destina a representar el exponente del número (supuesto en base 2), otra fracción al valor de la mantisa, y un último bit al signo. Los numerosos detalles que permiten que la representación sea utilizable se describen en profundidad en el estándar.

El problema que presenta la coma flotante es su gran complejidad. Las operaciones con números en coma flotante requieren multitud de pasos previos y posteriores de preparación y ajuste al estándar de representación, y eso se traduce en mucho más tiempo de cómputo si se realizan las operaciones en software, o en más tiempo, área y lógica si se realiza en hardware. La coma fija permite operar como si el número representado fuese un entero, de forma que no necesita hardware específico, y es significativamente más rápida. A cambio, su uso supone para el diseñador mucha más dificultad, siendo necesario un estudio de resolución para evaluar los efectos de la pérdida de precisión.

Debido a que la frecuencia de conmutación del convertidor conmutado que se modela en este TFG es de 200 kHz, y sabiendo que se requieren en torno a 100 ciclos de cálculo por ciclo de conmutación para que el modelo sea válido, es imprescindible el uso de señales de alta velocidad (de decenas de megahercios). La utilización del formato de coma fija para cumplir los requisitos de alta velocidad y frecuencia de cálculo se hace entonces ineludible. Se ha usado concretamente el estándar de coma fija con signo QX.Y, en el que la X indica el número de bits utilizados para representar la parte entera, sin incluir el bit más significativo con la información de signo, y la Y el número de bits de la parte decimal. La representación de éste estándar en el lenguaje VHDL, usado para el desarrollo del modelo, se explica en el CAPÍTULO 4.

CAPÍTULO 3: Desarrollo de un módulo de conversión

En el CAPÍTULO 2 se ha comentado cuál es la estructura de un sistema de control, y cómo el paradigma planta-regulador es sustituido a la hora de realizar pruebas por una nueva estructura HIL-regulador. También se ha explicado cómo dentro del modelo HIL aparecen dos partes muy diferenciadas, capaces al actuar conjuntamente de emular una planta analógica: un modelo digitalizado de ésta, generalmente sintetizado sobre una FPGA, y un circuito capaz de transformar las señales digitales de salida de ese modelo en señales analógicas similares a las que generaría una planta real.

Mientras que la forma en que se desarrolla un modelo digitalizado de una planta analógica (concretamente, de un convertidor DC-DC de tipo Buck multifase) es abordado en el siguiente capítulo, se desarrolla a continuación el proceso de diseño del circuito encargado de la conversión de señales.

El elemento clave del diseño que se emprende es un convertidor digital a analógico o DAC. Independientemente de su mecanismo de funcionamiento interno, un DAC toma como entrada un conjunto de señales que llevan información de carácter digital, y es capaz de proporcionar a la salida una señal analógica que lleva la información combinada de las señales digitales de entrada. En esencia, es eso lo que se busca que haga el circuito que se diseña, pero hay varias cuestiones adicionales a tener en cuenta.

Aquellos DACs que deben poder funcionar con altas tasas de muestreo suelen ofrecer la información analógica de su salida como una señal de corriente, y no de tensión. Dado que generalmente es más sencillo operar con tensión que con corriente, se hace imperativo colocar una posterior etapa de conversión de corriente a tensión.

Por regla general, el rango de la señal de salida no coincidirá con el rango de interés buscado, ni tendrá la potencia suficiente para ser útil. Es imprescindible amplificarla tratando de reducir al mínimo el ruido que aparece en ese proceso.

Adicionalmente, los propios fabricantes dan en las hojas de datos de sus componentes las directrices sobre a qué deben conectarse muchos de los pines del DAC. Condensadores externos desacoplan determinados pines de salida de tierra, resistencias conectadas entre ciertos pines sirven para ajustar la escala de la señal de información, y conectar algunos otros pines a tierra, a alimentación o a una determinada tensión externa sirve para ajustar el modo de funcionamiento, o la tensión de referencia con la que determinar si la señal digital de entrada es un uno o un cero en cada instante.

Por lo tanto, un proceso de acondicionamiento de la señal y de configuración del convertidor es imprescindible. Es el diseño de ese circuito lo que se ha abordado y se describe a continuación.

3.1. Estructura del diseño

El diseño por el que se opta consta de dos módulos fundamentales: el módulo de conversión de señales digitales a analógicas, con su desacoplo y ajuste de modo y escala; y el módulo de amplificación, que se conecta en cascada con el anterior y se encarga de transformar la corriente de salida de la etapa anterior a tensión y de amplificar ésta. Este

segundo módulo consiste esencialmente en un amplificador operacional colocado en una configuración que permita obtener una ganancia finita determinada, y su desacoplo.

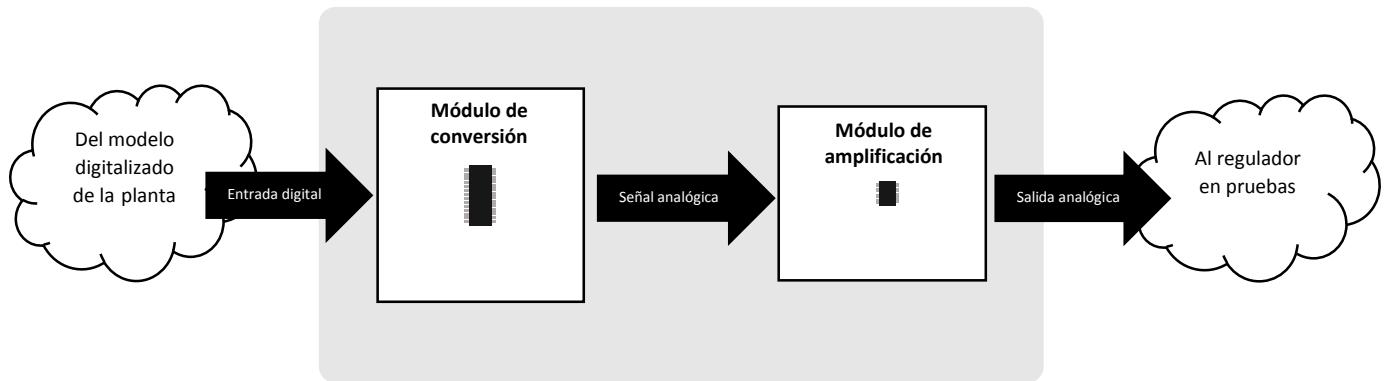


Figura 6. Estructura del circuito

Con el objetivo de poder evaluar por separado el rendimiento de los componentes, se opta en un primer momento por construir los módulos en placas separadas. De esta forma se pueden probar todas las combinaciones de DAC, amplificador y estructura de amplificación, y seleccionar la que ofrece mejores resultados.

Con posterioridad, y con el objetivo de hacer el diseño más compacto, se incluyen ambos en una única placa, con los componentes que hayan sido finalmente elegidos por sus mejores prestaciones.

3.2. Selección de componentes

En primer lugar se seleccionan posibles candidatos de conversor digital a analógico, en base a los requisitos especificados en el punto 3.2.2. Cuando se han seleccionado varios, se determina qué características debe tener el amplificador operacional de la etapa posterior de forma que pueda partirse de unas especificaciones generales para su elección. Una vez se dispone de una selección de DACs y amplificadores que cumplan los requisitos, puede determinarse qué otros componentes pasivos deben colocarse en el circuito, así como su valor y tolerancia.

3.2.1. Selección de DACs

Como se ha comentado, el DAC usado es la base del circuito. Su elección, por tanto, es crítica. Para conseguir un DAC lo más adecuado posible se sigue un proceso en varias etapas. En primer lugar, dada la vasta cantidad y variedad de componentes de este tipo que existen en el mercado, se establece cuáles deben ser los requisitos mínimos que deben cumplir los candidatos. Además, la búsqueda se restringe a los productos ofrecidos por los distribuidores Farnell [Farnell] o RS Components [RS] por asegurar la entrega en un tiempo reducido, y ofrecer la posibilidad de adquirir componentes de forma individual o en pequeñas cantidades.

Realizado el filtrado por especificaciones el número de DACs posibles disminuye pero sigue siendo muy alto. Con el objetivo de poder ordenarlos en función de sus características, atendiendo a lo favorables que son para la aplicación en cuestión, se crea una figura de mérito que ofrece una puntuación para cada componente.

Finalmente, de entre los DACs con mayor puntuación se selecciona un reducido número para ser comprados y poder someterlos a pruebas. Como se explica más adelante, se eligen cuatro candidatos que no coinciden exactamente con aquellos que más puntuación han obtenido en la figura de mérito.

3.2.1.1 Requisitos

Se han considerado como características indispensables de los candidatos:

- Tasa de muestreo mínima de 25 MSPS: alta velocidad. Lo ideal sería que se tomase una muestra en cada ciclo de reloj del modelo digital de la planta, de forma que no se pierda ninguna información. Una ventaja adicional sería que podría usarse el mismo reloj para el modelo digital y para la placa de traducción.
- Resolución de al menos 10 bits. Por regla general, los ADCs dedicados al control de convertidores conmutados tienen entre 8 y 12 bits de resolución. En consecuencia, la resolución del DAC de la placa que se diseña deberá ser igual o ligeramente superior con objetivo de no perder resolución, pero no debería ser muy superior por no suponer mejora alguna.
- Encapsulados de montaje superficial, por las mejores prestaciones que ofrecen. Quedan excluidos aquellos que no pueden soldarse a mano por la localización de sus pads, y que forzarían a adquirir un zócalo por separado (CLCC, PLCC, BGA...)
- Según se ha comentado, ofrecidos por Farnell o RS Components por la mayor facilidad de obtención en pequeñas cantidades y por el menor tiempo de espera.

Una vez filtrados los componentes siguiendo estos criterios, se llega a un total de 63 posibilidades, que sigue siendo un número muy elevado. Se hace necesaria una nueva etapa de selección.

3.2.1.2 Figura de mérito

Con el objetivo de obtener una puntuación para cada DAC que permita su ordenación en cuanto a su aptitud para la aplicación, se desarrolló la siguiente función:

$$Figura\ de\ mérito\ con\ precio = \begin{cases} 2^A * \frac{G}{H} & \text{si } A < 13 \\ \left(2^{12} + \frac{500 * (A-12) * \min(B,100) * C^3}{1000 + 100D * (1000F+1)} \right) * \frac{G}{H} & \text{si } A \geq 13 \end{cases}$$

$$Figura\ de\ mérito\ sin\ precio = \begin{cases} 2^A & \text{si } A < 13 \\ \left(2^{12} + \frac{500 * (A-12) * \min(B,100) * C^3}{1000 + 100D * (1000F+1)} \right) & \text{si } A \geq 13 \end{cases}$$

Siendo:

- A = Resolución en número de bits
- B = Tasa de muestreo en millones de muestras por segundo (MSPS)
- C = Pitch (Separación entre pines) en mm
- D = Error de escala completa
- F = Error de linealidad sobre 1 V
- G= Número de canales de conversión

- H= Precio en euros

La función se define de esa forma doble debido a que si la resolución es de 12 bits o menos (el valor mínimo es de 10 por los requisitos que se han comentado anteriormente), el resto de valores (a excepción del precio por canal) son mucho menos importantes, pues las aptitudes del componente serán previsiblemente peores. Si por el contrario la resolución es superior a 12, se considera que se ha pasado una barrera a partir de la cual no tiene sentido darle excesivo peso a cada bit adicional de resolución, de forma que el crecimiento pasa de ser exponencial a lineal.

Por otro lado, puede observarse que la influencia de la tasa de muestreo es limitada: una vez que se ha superado la barrera de los 100 MSPS resulta indiferente cuál es la tasa de muestreo máxima posible, pues en la aplicación para la que se selecciona el DAC no se superará una frecuencia de reloj de 100 MHz. Un muestreo más rápido que 100 millones de muestras por segundo no añadiría información.

El *pitch* o separación entre pines cobra un papel nada despreciable en la selección de componentes, dado que las soldaduras de componentes se van a realizar de forma manual. Encapsulados con un *pitch* menor a 1 mm presentan una importante dificultad para ser soldados sin errores, lo que hace que obtengan una puntuación baja y sean descartados a menos que presenten prestaciones muy superiores a las de sus competidores con mayores separaciones entre pines.

Los errores de linealidad y de escala completa, por el contrario, son mucho menos relevantes en la selección del DAC. Los factores que les acompañan sirven para que los valores obtenidos estén en el orden de magnitud de las unidades o decenas, de forma que su producto esté en varias centenas como máximo. Sumando 1000 a ese valor, se consigue que la influencia de estos parámetros sea reducida, y sirva únicamente para decidir en DACs muy parecidos entre sí en otros aspectos.

Finalmente, el parámetro de precio por canal permite tener en cuenta el precio del componente considerando cuántas conversiones simultáneas puede realizar a la vez. De esta forma se evita premiar a un componente A con menor precio unitario con un solo canal frente a otro B con dos canales algo más caro, si el precio de B es menor a dos veces el de A. Es en cualquier caso conveniente mencionar que el precio no es un parámetro que deba tener gran relevancia en la elección de componente. Dejando de lado los precios significativamente mayores que tienen aquellos DACs con multitud de prestaciones adicionales que se desperdiciarían en este diseño, el orden de magnitud del coste de los conversores es el mismo. Debido a que lo que se busca es un prototipo, y no un modelo comercial en el que ajustar el precio al céntimo para maximizar el beneficio es el objetivo, se ha optado por dejar el precio en segundo plano elaborando una figura de mérito adicional en el que el factor de precio por canal no se tiene en cuenta. Es significativo, sin embargo, que aquellos 5 componentes que mayor puntuación obtienen en la figura de mérito que tiene en cuenta el precio coinciden con los que tienen la máxima puntuación en la figura de mérito sin precio.

Para llevar a la práctica el sistema de ordenación antes mencionado, se crea una hoja Excel en la que se introducen manualmente todos los parámetros de interés de los 63 DACs que quedaron tras el filtrado por especificaciones. Una vez implementada la fórmula, se puede ordenar el conjunto por puntuación, consiguiéndose un *ranking* de componentes. En la Tabla 1 y Tabla 2 a continuación se muestran los 15 primeros de la clasificación:

Nombre	Resolucion	Tasa (MSPS)	Pitch (mm)	Error FSR (%)	Error linealidad (LSB)
AD9754ARZ	14	125	1,27	5	3
AD9764ARZ	14	125	1,27	7	6,5
AD9752ARZ	12	125	1,27	5	2
DAC904U	14	200	1,27	10	3
DAC904UG4	14	200	1,27	10	3
AD768ARZ	16	40	1,27	1	8
DAC902U	12	200	1,27	10	2,5
AD9762ARZ	12	125	1,27	10	4
AD9740ARZ	10	210	1,27	2	0,7
THS5651AIDW	10	125	1,27	2,3	1
AD9744ARUZ	14	210	0,65	0,5	5
AD9742ARUZ	12	210	0,65	0,5	2,5
AD9754ARUZ	14	125	0,65	5	3
LTC1668CG#PBF	16	50	0,65	0,0061	8

Tabla 1: Tabla de DACs candidatos (I)

Nombre	Error linealidad sobre 1 V	Canales	Precio	Figura merito sin precio	Figura mérito con precio
AD9754ARZ	0,000183105	1	12,25	655,8726923	53,54062794
AD9764ARZ	0,000396729	1	12,88	527,8104461	40,9790719
AD9752ARZ	0,000488281	1	10	481,0493287	48,10493287
DAC904U	0,000183105	1	17,11	478,1518766	27,9457555
DAC904UG4	0,000183105	1	18,23	478,1518766	26,22884677
AD768ARZ	0,00012207	1	37,19	449,0869925	12,07547708
DAC902U	0,000610352	1	16,72	321,4194168	19,22364933
AD9762ARZ	0,000976563	1	10,84	281,8747051	26,00320157
AD9740ARZ	0,000683594	1	7,83	156,9173913	20,04053529
THS5651AIDW	0,000976563	1	9,86	144,1998263	14,62472883
AD9744ARUZ	0,000305176	1	12,06	131,3754943	10,89349041
AD9742ARUZ	0,000610352	1	9,76	104,1041833	10,66641222
AD9754ARUZ	0,000183105	1	12,25	87,93230471	7,178147323
TC1668CG#PBF	0,00012207	1	25,53	83,64844574	3,276476527

Tabla 2: Tabla de DACs candidatos (II)

3.2.1.3 Selección final de conversores

Una vez se dispone de un conjunto reducido de posibilidades donde elegir, es más sencillo tener en cuenta criterios adicionales a los estrictamente analíticos que se han aplicado hasta ahora.

En primer lugar, en la selección final se ha tratado de que las familias de los candidatos sean diferentes, de forma que se evite la posibilidad de algún fallo compartido por todos. Adicionalmente, en lugar de tratar de que todos tengan la máxima resolución posible, se buscan opciones con resoluciones diferentes, para poder discernir si la resolución es, a partir de un mínimo, relevante en la aplicación.

Tras todo el proceso se llega a la selección final de conversores. Se decide adquirir:

- Un DAC modelo AD9754ARZ, de Analog Devices [AD9754ARZ]
- Un DAC modelo AD9752ARZ, de Analog Devices [AD9752ARZ]
- Un DAC modelo DAC904U, de Texas Instruments [DAC904U]
- Un DAC modelo THS5651AIDW, de Texas Instruments [THS5651AIDW]

Sus características pueden contemplarse en la Tabla 3 y en la Tabla 4

Nombre	Resolucion	Tasa (MSPS)	Pitch (mm)	Error FSR (%)	Error linealidad (LSB)
AD9754ARZ	14	125	1,27	5	3
AD9752ARZ	12	125	1,27	5	2
DAC904U	14	200	1,27	10	3
THS5651AIDW	10	125	1,27	2,3	1

Tabla 3: Características DACs seleccionados (I)

Nombre	Error linealidad sobre 1 V	Canales	Precio	Figura merito sin precio	Figura mérito con precio
AD9754ARZ	0,000183105	1	12,25	655,8726923	53,54062794
AD9752ARZ	0,000488281	1	10	481,0493287	48,10493287
DAC904U	0,000183105	1	17,11	478,1518766	27,9457555
THS5651AIDW	0,000976563	1	9,86	144,1998263	14,62472883

Tabla 4: Características de los DACs seleccionados (II)

Una ventaja adicional que ofrece la selección anterior es que todos los DACs seleccionados comparten encapsulado y *pinout*, es decir, todos ellos tienen el mismo número de pines y los pines que ocupan la misma posición desempeñan la misma función. Hay una sola excepción a lo anterior: el DAC THS5651AIDW, que además de tener 10 bits, por lo que 4 de sus pines no se conectan a nada, requiere de un condensador de desacoplo adicional en uno de sus pines, según indica el fabricante en su hoja de datos.

Esta coincidencia de huellas permite que el circuito de acondicionamiento a desarrollar sea el mismo, o muy similar, en todos los casos, permitiendo reducir el tiempo dedicado a su diseño.

3.2.2. Selección de amplificadores operacionales

Una vez se han elegido DACs, se pueden conocer las características de la señal de salida de la etapa de traducción, lo que permite determinar cuáles son los requisitos que deberán cumplir los amplificadores candidatos.

El número de amplificadores de alta velocidad y gran ancho de banda candidatos es mucho más reducido que el de DACs de la selección anterior. Adicionalmente, las características de cada amplificador son más difíciles de cuantificar que en el caso de los DACs, por ser componentes analógicos. Ambos factores sirven para explicar el que en esta selección no se utilizase una nueva figura de mérito.

Un dato importante con el que se cuenta hecha la selección de los DACs es que la corriente máxima de salida de la etapa de traducción es de 20 mA, como se comentará más adelante. Para traducir esa corriente a tensión, una de las estructuras que se diseña es la de un amplificador no inversor. La tensión a amplificar se obtiene en ese caso haciendo circular la corriente de salida de la etapa de traducción por una resistencia de 50 Ω , de forma que la caída de tensión en ella es de 1 V.

3.2.2.1 Requisitos

Se ha considerado imprescindible:

- Que los candidatos puedan ser alimentados con una tensión única y tierra, en lugar de requerir una tensión positiva (+Vcc), la misma tensión negativa (-Vcc) y tierra. Si además la tensión de alimentación requerida es positiva, puede usarse la misma alimentación para la etapa de traducción y la de amplificación.
- Que los candidatos sean *rail to rail*, es decir, que sean capaces de ofrecer a su salida un rango de tensiones que vaya desde 0 V (tierra) hasta la tensión de alimentación, sin quedar limitada la tensión máxima de salida a una fracción de la de alimentación.
- Que sean de alta velocidad y gran ancho de banda, pudiendo amplificar las señales que provienen de la etapa anterior.

Se ha comentado ya que el diseño está pensado para que pueda traducir señales procedentes de modelos con un reloj de 100 MHz como máximo. Esto significa que la tasa de muestreo de la etapa anterior será como máximo de 100 MSPS, y por tanto según el teorema de Nyquist, la frecuencia máxima de las señales de entrada a la etapa de amplificación será de 50 MHz.

Suponiendo un amplificador *rail to rail*, una etapa de conversión y de amplificación que comparten una alimentación positiva de 5V, y con el dato que se mencionó anteriormente sobre una tensión a amplificar de 1V, se busca que la ganancia mínima del amplificador realimentado sea de 5, de forma que se aproveche el rango completo de posibles tensiones de salida.

Con los valores máximos de ganancia y ancho de banda necesarios, podemos establecer su producto en $5 \cdot 50\text{MHz} = 250 \text{ MHz}$.

Se sigue este proceso porque ganancia y ancho de banda de un amplificador no son independientes, de forma que el dato que ofrecen los fabricantes es precisamente su producto.

- Que sólo tengan un canal de amplificación por componente. Debido a que los DACs elegidos sólo tienen un canal de conversión, usar amplificadores con varios canales significaría desperdiciar uno de ellos.
- Encapsulados de montaje *thru-hole* o de montaje superficial, excluyendo aquellos que no pueden soldarse manualmente dada la disposición de sus pads (BGA, PLCC...). *Pitch* de al menos 1 mm.

3.2.2.2 Selección final de amplificadores

Tras realizar el filtrado por especificaciones, se llega a un conjunto reducido de posibilidades. Finalmente, se seleccionan los modelos:

- CLC1005 en encapsulado SOIC de 8 pines, como opción de bajo coste. [CLC1005]
- ADA4857 también en encapsulado SOIC de 8 pines. En teoría, se trata de la opción que más ancho de banda ofrece, además de ser la que menor ruido produce. [ADA4857]

Al igual que ocurría con los convertidores seleccionados, el hecho de que los operacionales elegidos compartan encapsulado y *pinout* facilita el diseño del circuito del que deben formar parte.

Las características más relevantes de los candidatos pueden observarse en la Tabla 5 a continuación.

Modelo	Ganancia en lazo abierto	Producto ancho de banda-ganancia	Slew Rate	Encapsulado	Pitch	Rango de alimentación	Precio
CLC1005	78 dB	260 MHz	145 V/ μ s	SOIC-8	1,27 mm	2,7 a 5,5 V	0,319 €
ADA4857	57 dB	750 MHz	2800 V/ μ s	SOIC-8	1,27 mm	5 a 10 V	3,630 €

Tabla 5: Características de los amplificadores operacionales

3.2.3. Otros componentes

En este punto, se hace imprescindible tener una idea clara de la estructura que va a tener cada módulo, de forma que pueda seleccionarse el conjunto de componentes pasivos que más se adecúe. Aunque en el siguiente apartado se hablará más extensamente sobre la estructura de cada circuito, una breve descripción de los componentes requeridos en cada parte es necesaria en este punto.

A grandes rasgos, el módulo de traducción, además de un DAC, únicamente requiere de condensadores de desacoplo de valor 100 nF, según lo indicado por el fabricante, y de una resistencia que permita ajustar el valor de la corriente de referencia. Cuando se realizan los cálculos que permiten determinar el valor de esa resistencia (ver apartado 3.3.1), se obtiene un valor de 1984 Ω .

Ante la tremenda dificultad para conseguir ese valor, se decide comprar un lote de 100 resistencias de 2 k Ω con tolerancias del 5%, y seleccionar aquellas cuyo valor, medido con un multímetro, se acerque más al requerido. Se selecciona un encapsulado de montaje superficial tipo 1206, que puede soldarse manualmente sin excesivos problemas.

En cuanto a los condensadores, el valor de 100 nF especificado por el fabricante es común, y la exactitud en su valor no es crítica. Así pues, se selecciona un lote con 250 condensadores de ese valor, de montaje superficial con huella 1206 y tolerancia del 10%.

En el caso del módulo de amplificación se busca probar la eficacia de dos configuraciones diferentes: una inversora y otra no inversora. Además de condensadores de desacoplo, papel que pueden desempeñar los que se han adquirido para la etapa de traducción, se requieren resistencias para el ajuste de la ganancia y condensadores que filtren el ruido de muy alta frecuencia.

Según los cálculos que se muestran en el apartado 3.3.2, para la configuración inversora se requiere una resistencia de ajuste de ganancia de valor 250 Ω . Se elige el valor comercial más próximo: 240 Ω . Dado que se trata de una resistencia cuyo valor es crítico, se escoge un modelo con una tolerancia del 1%, adquiriéndose 50 unidades de huella 1206.

Por otro lado, para la opción que representa una configuración de amplificación no inversora, se requieren varias resistencias adicionales para ajustar la ganancia y convertir la corriente de entrada en tensión a amplificar. Se determina finalmente la posibilidad de usar resistencias de valor 100 Ω , 1k Ω , 4k7 Ω y 27 k Ω . A tal fin, se adquiere un pack de resistencias de montaje superficial y valores comerciales, con huella de tipo 1206 y tolerancia del 1%.

Los condensadores de reducción de ruido han de ser de pequeño valor, pues el ruido a filtrar es de muy alta frecuencia. Se eligen condensadores de 1 pF con una tolerancia del 25%, y de 10 pF con 5% de tolerancia, ambos en huella 0805. Se usarán los mismos tanto en la configuración inversora como no inversora.

Adicionalmente, y por motivos que se comentan más adelante, se hace necesario que en las placas conjuntas exista una salida cuyo rango varíe entre 0 y 1 V, en lugar de entre 0 y 5 V. Para ello se hace uso de un simple divisor de tensión con respecto a la salida normal, usándose resistencias de 680 y 2k7 Ω del pack mencionado. También se añaden condensadores de valor 1 μ F y montaje superficial en huella 1206 para completar la red de desacoplo de las placas conjuntas.

Finalmente, se adquieren todos aquellos conectores que permiten la interconexión de los módulos en desarrollo entre sí y con el exterior: tiras de pines hembra para introducir a los circuitos señales o alimentación, y tiras de pines macho para facilitar la medida de señales en puntos de test (*testpoints*) del circuito. Se adquieren asimismo conectores IDC macho y hembra de 24 pines por los motivos comentados en el apartado 3.4.1, y cinta plana de 24 pistas.

Como resumen, se adquieren los siguientes componentes:

- Resistencias modelo CRCW1206240RFKEA, de valor 240 Ω , montaje superficial con huella 1206 y tolerancia 1%.
- Resistencias modelo ERJT08J202V, de valor 2 k Ω , montaje superficial con huella 1206 y tolerancia 5%.
- Condensadores modelo C0805CGACTU, de valor 1 pF, montaje superficial con huella 0805 y tolerancia 25%.
- Condensadores modelo GRM2165C1H100JZ01D, de valor 10 pF, montaje superficial con huella 0805 y tolerancia 5%.
- Condensadores modelo CL31B104KBCNNL, de valor 100 nF, montaje superficial con huella 1206 y tolerancia 10%.
- Condensadores modelo C1206X105K3RACTU, de valor 1 μ F, montaje superficial con huella 1206 y tolerancia 10%.
- Pack de resistencias de montaje superficial de valores comerciales, en huella 1206 y 1% de tolerancia. Se usan resistencias de valor 100 Ω , 680 Ω , 2k7 Ω , 4k7 Ω y 27 k Ω .
- Tiras de pines macho y hembra.
- Conectores IDC macho y hembra de 24 pines.
- Cinta plana de 24 pistas.

3.3. Diseño y evaluación de placas separadas

Una vez seleccionados los componentes, puede comenzarse el desarrollo de los circuitos impresos donde serán soldados. Para este proceso, se parte de un esquemático en papel basado en las instrucciones de los fabricantes de los componentes, y en la funcionalidad buscada.

A partir del borrador del esquemático, se usa el software *Altium Designer*, para diseño electrónico. La herramienta permite realizar el esquemático en formato digital, y a partir de las huellas que se han introducido para cada componente del circuito, es capaz de generar un PCB en el que aparecen las huellas de los componentes y las interconexiones que deben realizarse

según se indicaban en el esquemático. El usuario puede entonces colocar cada componente en la posición que desee, y dibujar las pistas que los conectan, generando después los ficheros (*Gerber* y *NCdrill*) que permitirán la construcción física del PCB al ser introducidos en la fresadora de la escuela. Los esquemáticos completos y la colocación de componentes y rutado (*layout*) en *Altium* se muestran en detalle en el ANEXO A.

3.3.1. Placa de conversión

Los fabricantes de DACs indican, para todos los candidatos seleccionados, la necesidad de configurar una determinada corriente de referencia interna para el DAC. El funcionamiento de éste se basa en sumar más o menos veces, según los bits de la señal digital de entrada, esa corriente, de forma que el valor de la corriente de escala completa de salida, que en todos los casos es ajustable entre 2 y 20 mA, es igual a 32 veces el de esa corriente de referencia. El esquema es idéntico en todos los DACs seleccionados: el valor de la resistencia de ajuste se obtiene dividiendo la tensión de referencia interna de los DACs, en todos los casos igual a 1,24 V, entre la corriente de referencia deseada. Como buscamos un rango dinámico máximo, elegimos una corriente de escala completa de 20 mA, y entonces:

$$R = \frac{1,24 \text{ V}}{\frac{20 \text{ mA}}{32}} = 1984 \Omega$$

La resistencia de 2 kΩ que finalmente se usa provoca que la corriente de escala completa sea finalmente de 19,84 mA en lugar de 20, pero es una diferencia que se considerará despreciable.

Dado que los cuatro candidatos a conversor presentan un *pinout* prácticamente idéntico, puede diseñarse una única placa en la que se suelde uno u otro según corresponda. La excepción a esto es el DAC THS5651AIDW como se comentó en el epígrafe 3.2.1.3, que requiere de un condensador de desacoplo adicional entre su pin número 19 y la alimentación, por lo que se realiza una pequeña modificación en el circuito.

Es importante destacar que los DACs elegidos permiten obtener la información de forma diferencial, pues ofrece a su salida tanto la corriente que lleva la información analógica, como su negada. Colocar un amplificador diferencial en la siguiente etapa permitiría hacer uso de las dos corrientes, pero por sencillez en el diseño, se decide que la etapa de amplificación sea *single ended*, y en consecuencia el pin por el que se ofrece la corriente negada se lleva a tierra según indican los fabricantes (ver Figura 7 a continuación).

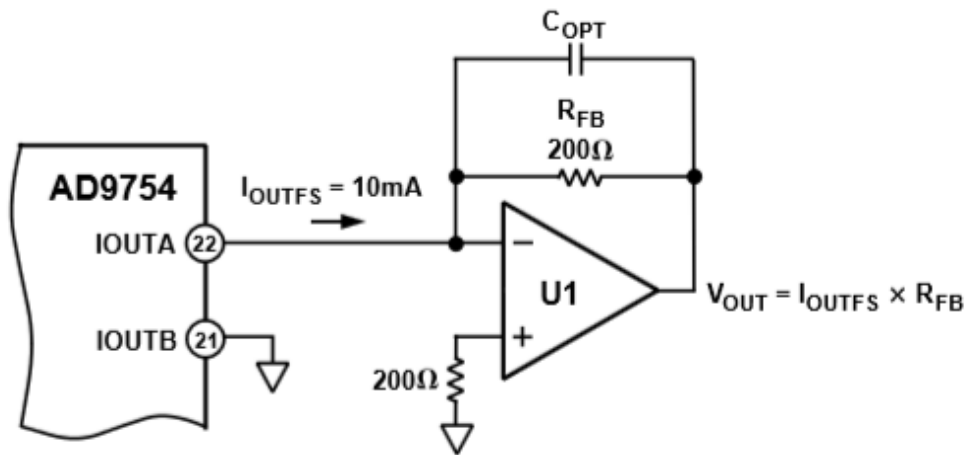


Figura 7: Configuración de salida single-ended típica

Una cuestión adicional se refiere a las tensiones de referencia de tierra, y a las alimentaciones. Los DACs elegidos dan la opción de usar una tensión de referencia de tierra diferente para la parte analógica y digital. Por sencillez en el diseño y sobre todo en el rutado, se ha optado por unir ambas tierras, lo cual es además recomendable para reducir ruidos según [Kester]. Con respecto a las alimentaciones, todos los DACs permiten alimentar de forma independiente la parte analógica y digital. El diseño se realiza pensando en una tensión de alimentación para la etapa analógica de 5 V, tensión a la que los DACs ofrecen mejores prestaciones según los fabricantes. Por otro lado, en la etapa digital se permitirán alimentaciones en el rango entre los 3.3 V y los 5 V.

3.3.1.1 Circuito común

Cómo se ha comentado, para los DACs AD9754ARZ, AD9752ARZ y DAC904U se ha diseñado una única placa común. Dado que el número de bits de resolución entre el AD9752ARZ y los otros dos es diferente, simplemente quedarán dos pines con sus correspondientes pistas hasta el conversor a los que no habrá que conectar ningún cable cuando el DAC soldado sea el AD9752ARZ. El esquemático que se usa en la placa común se muestra en la Figura 8. Aunque el componente mostrado es el AD9754, el pinout es común.

Se aprecia en la figura que el pin 16 está conectado a tierra (*AnalogGND*) para que se use la referencia de tensión interna, según indica el fabricante. En consecuencia, el pin 17, por donde se introduciría esa referencia, está desacoplado de tierra.

El pin 14, por su parte, sirve para colocar el DAC en modo de ahorro de energía, opción que no interesa en la aplicación. Una conexión *pull-down* interna permite que pueda dejarse sin conectar.

Puede consultarse en el ANEXO A una descripción más pormenorizada de la función de cada pin según es ofrecido por el fabricante, así como la huella de cada uno de los DACs.

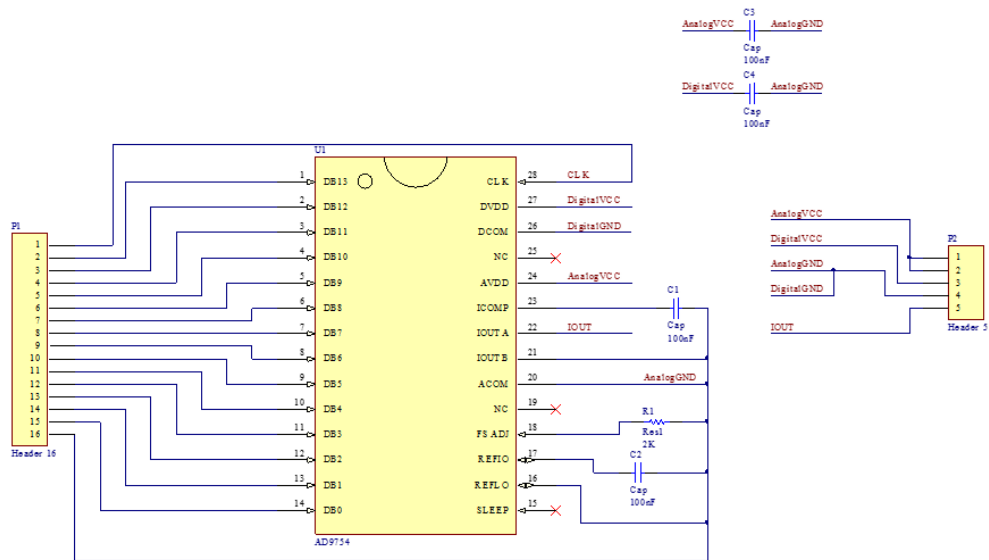


Figura 8: Esquemático de la placa de conversión para el AD9754ARZ

3.3.1.2 Circuito del THS5651AIDW

El circuito en el que se colocará el THS5651AIDW es diferente, pero muy sutilmente. La única diferencia es la necesidad de desacoplar el pin 19 de la alimentación analógica a través de un condensador de 100 nF adicional, como puede verse en el esquemático de la Figura 9. En la práctica, incluir este condensador modifica ligeramente la posición del resto de componentes en el PCB. Además, debido a que el circuito diseñado es exclusivo para el DAC en cuestión, que tiene 10 bits, se colocan únicamente 10 pistas que unen el conversor con el conector de entrada, en lugar de colocar 14 y dejar cables sin conectar como se hace en la placa que comparte el resto de DACs.

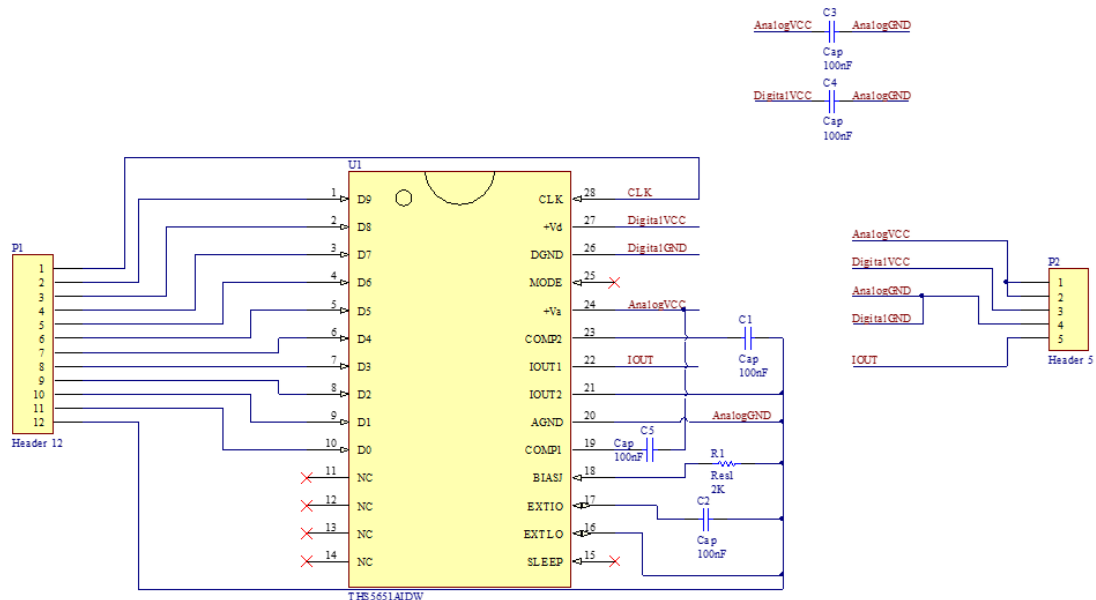


Figura 9: Esquemático de la placa de conversión para el AD9754ARZ

3.3.2. Placa de amplificación

Según se ha explicado en el apartado 3.3.1, la salida de la etapa de conversión es una corriente única. El circuito de la etapa de amplificación sirve por tanto para transformar esa corriente en tensión, y amplificarla.

Las formas de hacer esto son múltiples, pero se van a probar dos configuraciones que destacan por su sencillez. La primera, casi idéntica a la mostrada en la figura Figura 7, es una configuración inversora, por lo que la tensión de salida será negativa, y en consecuencia requerirá de una alimentación también negativa. La necesidad de incluir una etapa adicional de inversión si se desea una tensión de salida del sistema positiva supone una importante desventaja.

La segunda configuración se muestra en la Figura 13, y es la formada por una resistencia que actúa de conversor corriente a tensión, y un amplificador no inversor que amplifica el voltaje que aparece en dicha resistencia cuando la atraviesa la corriente de entrada.

Ambas configuraciones se desarrollan en más detalle a continuación.

3.3.2.1 Configuración inversora

Si se opta por una etapa inversora, se usará un circuito de las características del mostrado en la Figura 10:

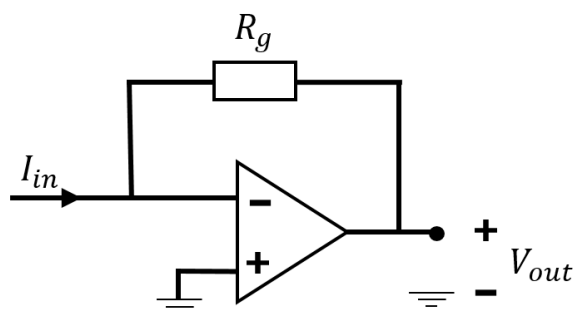


Figura 10: Conversor de corriente a tensión

Es sencillo observar que la tensión de salida está gobernada por la tensión de entrada según la ecuación

$$V_{out} = -R_g \cdot I_{in}$$

Lo primero que se pone de manifiesto es que para una corriente de entrada en el sentido de la mostrada en el dibujo, la tensión de salida es negativa. Debido a que en el diseño que se plantea la corriente es *saliente* de la etapa anterior, se está en ese caso. En consecuencia, el operacional deberá ser alimentado con una tensión negativa, diferente a la de la etapa de conversión, lo cual es una importante desventaja. Como además se desea que el diseño ofrezca tensiones positivas a su salida, sería necesaria una etapa adicional de inversión, formada por otro operacional alimentado con tensión positiva en una configuración como la mostrada en la Figura 11.

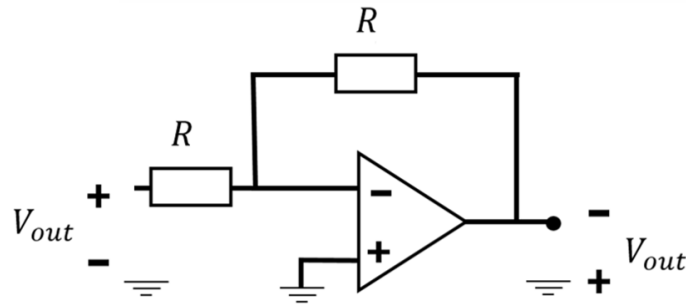


Figura 11: Inversor simple con operacional

En cualquier caso, se fija el valor de la alimentación en - 5 V. El máximo valor de corriente de entrada a la etapa (20 mA por los motivos indicados en el epígrafe 3.3.1) deberá corresponder con la tensión de salida más alta en magnitud de forma que no se desperdicie ningún fragmento del rango de tensiones que puede ofrecer un operacional *rail to rail* alimentado con -5 V y masa. El valor de esta tensión de escala completa será entonces igual al de la tensión de alimentación: -5 V.

La resistencia de ajuste de ganancia R_g se obtiene despejando: sabiendo que la tensión de salida será de -5V cuando la corriente de entrada sea de 20 mA, se obtiene un valor para R_g de 250 Ω . Como ya se ha indicado en el apartado 3.2.3, ese valor no es comercial, y se opta por usar una resistencia de 240 Ω , más sencilla de obtener. En consecuencia, la tensión de salida máxima es de -4,8 V, no aprovechándose totalmente el rango de tensiones posibles.

Se muestra a continuación (Figura 12) el esquemático de este circuito. Si se quiere conocer la huella y la función de cada pin del amplificador operacional, puede consultarse el ANEXO A.

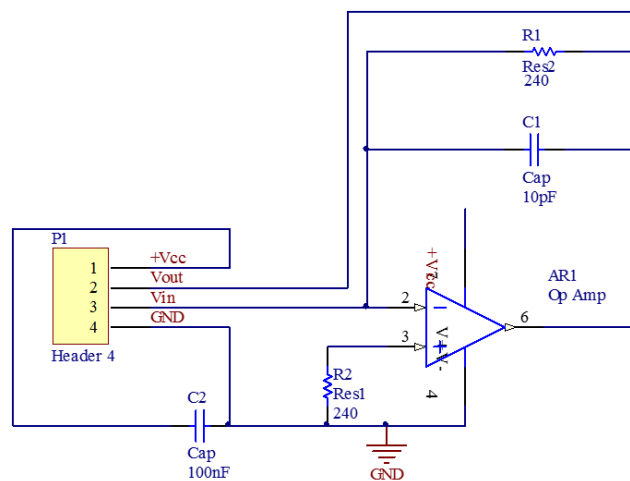


Figura 12: Etapa de amplificación en configuración inversora

Se usan los mismos condensadores que en la etapa de conversión para la red de desacoplo. Además, se añade la huella de un condensador en formato 0805 para ser soldado en paralelo con la resistencia de ajuste de ganancia (el *layout* del PCB diseñado se encuentra en el ANEXO A). Esta disposición hace que el conjunto de condensador y resistencia se comporte como un filtro paso bajo de frecuencia de corte $1/RC$. Soldando el condensador de 10 pF, o el de 1 pF puede controlarse el orden de magnitud de dicha frecuencia de corte, y por tanto la frecuencia del ruido que se filtra. En cualquier caso, cualquiera de las dos opciones de

condensador dan frecuencias de corte superiores a la frecuencia de reloj máxima para la que se está diseñando el sistema (100 MHz), así que no se espera que la señal principal se vea afectada.

La resistencia R_2 del esquemático no es imprescindible para el funcionamiento de la etapa. Sin embargo, sirve para que la corriente de offset del operacional se anule, reduciéndose de ésta forma las no idealidades que puedan perturbar el funcionamiento esperado.

3.3.2.2 Configuración no inversora

Usando una configuración no inversora el número de resistencias de la etapa de amplificación aumenta, como puede observarse en la Figura 13. Sin embargo, si se tiene en cuenta que una configuración inversora requiere de una etapa adicional de inversión para conseguir una salida positiva, la configuración no inversora requiere un amplificador menos, reduciéndose ruidos, consumo y otros problemas.

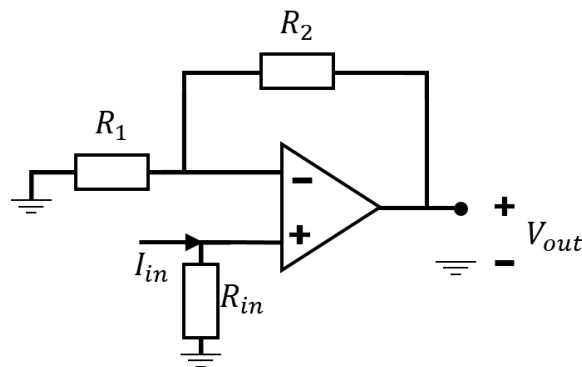


Figura 13: Circuito conversor corriente a tensión, con amplificador no inversor

Una ventaja adicional es que puede alimentarse con la misma tensión de 5 V que se usaba para alimentar la parte analógica etapa de conversión, de forma que sólo sea necesaria una alimentación en todo el diseño.

Una configuración no inversora habitual es la formada por un circuito como el de la Figura 13, pero eliminando la resistencia de entrada R_{in} . Suponiendo que la impedancia de entrada de los terminales del operacional es muy alta, la corriente de entrada sería despreciable, de forma que se amplificaría la tensión que llegase al terminal no inversor.

En este diseño, sin embargo, la etapa anterior no proporciona una señal de tensión sino de corriente. En consecuencia, primero hay que obtener una tensión proporcional a la corriente, y posteriormente amplificarla. Debido a que la impedancia de entrada del terminal no inversor del operacional es enormemente mayor que la resistencia de entrada, tenemos que:

$$V_+ = I_{in} \cdot R_{in}$$

Habiendo conseguido de esta manera una tensión que se puede amplificar a través de la resistencia de entrada.

Por otro lado, en el circuito se cumple:

$$V_{out} = \left(1 + \frac{R2}{R1}\right) \cdot V_+$$

De esta forma se puede ajustar la ganancia de tensión del circuito, y en consecuencia el rango de variación de la señal de salida. Considerando la corriente de entrada de valor igual a 20 mA, valor justificado en el apartado 3.3.1, y colocando una resistencia de entrada de 50 Ω que no representa una carga demasiado elevada para el DAC de la etapa anterior, la caída de tensión será de 1 V. Si deseamos aprovechar al máximo el rango de tensiones de salida del operacional alimentado por 5 V, la ganancia deberá ser de 5. En esas condiciones, el cociente $\frac{R2}{R1}$ deberá ser igual a 4.

Conseguir esa relación con resistencias de valores comerciales es algo complejo, pero puede conseguirse utilizando una resistencia de 1 k Ω como R1, y el paralelo de una resistencia de 4k7 Ω y una de 27 k Ω , con valor 4003 Ω , como R2. Por otro lado, usando dos resistencias de 100 Ω en paralelo para obtener los 50 Ω requeridos de resistencia de entrada, quedan determinados los valores de todas las resistencias necesarias según se indicaron en el apartado 3.2.3.

El esquemático definitivo puede observarse en la Figura 14.

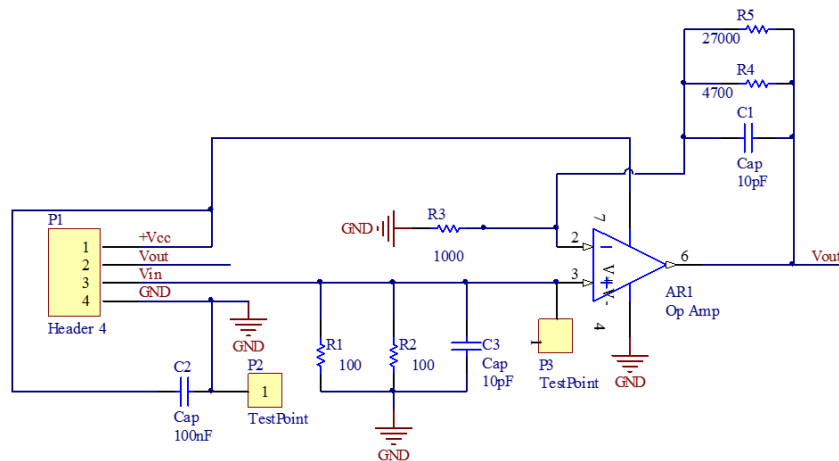


Figura 14: Etapa de amplificación en configuración no inversora

De forma similar a lo que se ha hecho en la etapa inversora, se incluyen condensadores en paralelo con las resistencias de entrada y con las de ajuste de ganancia para reducir posibles ruidos de alta frecuencia. Los condensadores de desacoplo de valor 100 nF permiten que las inductancias parásitas de la red de alimentación no introduzcan perturbaciones en las señales. Se han añadido además puntos de medición (*testpoints*) de forma que pueda comprobarse el correcto funcionamiento del circuito observando si las señales que llegan a cada punto son las adecuadas.

3.3.3. Pruebas

Con el objetivo de determinar el rendimiento de cada una de las posibilidades (DACs, operacionales y estructura de amplificación), se diseña una batería de pruebas a las que someter a los circuitos, una vez construidos. Las pruebas consisten en crear unos generadores de señales digitales de determinadas características, con las que se pueda atacar a los circuitos creados de conversión y amplificación. A través de un osciloscopio con capacidad para realizar

la transformada rápida de Fourier (FFT), puede verse la forma de la señal analógica resultante en el tiempo, y las características de su espectro (armónicos, ancho de banda...).

Con los resultados de las pruebas, puede determinarse cuál es la posibilidad que mejores prestaciones ofrece, de forma que sea la que se seleccione para la construcción de la placa conjunta.

3.3.3.1 Creación de generadores de señales

Los generadores de señales se describen en lenguaje VHDL, de forma que sean sintetizables en una FPGA, que es la que se conectará al PCB que se está diseñando. Se trata simplemente de circuitos digitales que poseen 16 puertos de salida para bits digitales, y un puerto de reloj. Para facilitar el desarrollo del código VHDL y aumentar la velocidad, en lugar de generar las distintas señales de diferente manera en el hardware, se leerán de una memoria RAM embebida en la FPGA y configurada previamente. La programación de dicha memoria se realiza copiando en su modelo VHDL el código que genera un script desarrollado en MATLAB a tal fin.

De esta manera, la función a codificar se genera en un script MATLAB, y éste, una vez ejecutado, genera un código hexadecimal que, al ser copiado en el lugar indicado del modelo VHDL de la memoria, permite programar la misma. El resto del código VHDL se limita a leer secuencialmente la memoria, y enviar los valores encontrados a la salida:



Figura 15: Generadores de señales

Para generar el código hexadecimal de la memoria, el script MATLAB parte de una definición de la señal que se quiere generar, por ejemplo $\sin\left(\frac{2*\pi}{T} * t\right)$. Una vez se tiene la definición de la señal, uno a uno sus valores se convierten a formato en coma fija, previa especificación del número de bits de parte entera y decimal. Los valores se llevan a un fichero de texto en el que se respeta además el formato exacto para la configuración de la memoria.

El código de configuración generado en el fichero de texto se copia en el modelo VHDL de la memoria. Dicho modelo es en realidad una descripción de entidad de la memoria, y no una descripción funcional, pues la memoria es un módulo completo embebido en la FPGA. Se muestran únicamente sus puertos, de forma que pueda ser instanciada en alguna estructura jerárquica, y se ofrece además un espacio donde insertar el código de configuración. Para obtener el *template* de la memoria usada, donde puede insertarse el código de configuración, puede accederse en el editor de Xilinx ISE a: Edit -> Language Templates -> VHDL -> Device Primitive Instantiation -> Spartan 3 -> RAM/ROM -> Block RAM -> Single port -> 1k x 16 +2 Parity Bits (RAMB16_S18).

Con la memoria configurada, desde un módulo *top-level* únicamente resta instanciar la memoria y leer secuencialmente todos sus valores, volviendo a empezar al llegar al final (de no ser así, la duración de la señal sería excesivamente breve). Para evitar transiciones indeseadas, la señal generada debe ser periódica y debe leerse de memoria un número entero de períodos. Asignando los valores leídos a señales que se asignan a pines de salida de la FPGA en el archivo UCF, se consigue el objetivo buscado: un generador de señales. El reloj del modelo también debe sacarse por uno de los pines de forma que pueda llevarse al circuito conversor diseñado.

Puede consultarse el código VHDL de los generadores de señales en el ANEXO A. También se muestra el fichero UCF donde se asignan las señales de salida del generador a los pines de la FPGA. La asignación se ha realizado teniendo en cuenta la disposición de pines de la placa Spartan-3 Starter Kit Board.

3.3.3.2 Señales utilizadas

Se han creado cinco señales con características diferentes.

La primera de ellas no es generada por un módulo como el descrito en el apartado anterior. Se trata de un generador de señales escalón, controlado por los switches de que dispone la placa Starter Kit Board. De una señal digital de 16 bits, los 8 bits menos significativos se dejan a 0, y los 8 más significativos se controlan con los 8 switches de que dispone ésta placa. Puede evaluarse de esta forma el comportamiento en régimen permanente del circuito de conversión en diseño. El código VHDL de este generador de señales escalón puede encontrarse en el ANEXO A.

El resto de señales sí se generan según se ha explicado en el epígrafe 3.3.3.1. Son las siguientes:

- Una señal sinusoidal de frecuencia 50 kHz, para comprobar el funcionamiento del sistema con un tono lo más puro posible, de baja frecuencia.
- Una señal triangular de frecuencia 50 kHz, para evaluar el funcionamiento con una señal de banda más ancha que un tono, y de baja frecuencia.
- Una señal sinusoidal de frecuencia 1 MHz, para observar los problemas que puedan surgir con tonos de alta frecuencia.
- Una señal triangular de frecuencia 1 MHz, para probar al sistema con una señal de alta frecuencia y banda más ancha que un tono.

Para generar las señales de baja frecuencia (50 kHz), se utiliza el reloj de 50 MHz que se encuentra en la Starter Kit Board. Por el contrario, a la hora de generar las señales de alta frecuencia (1 MHz) se aumenta la frecuencia de reloj a 100 MHz, para observar el comportamiento de los DACs. Este reloj se sintetiza a partir del de 50 MHz mencionado antes, y un DCM capaz de multiplicar la frecuencia por dos. Una explicación más detallada sobre la utilización del DCM se da en el CAPÍTULO 4.

3.3.4. Resultados

Una vez creados los generadores de señales, implementados en una FPGA, y conectados con el circuito de conversión, puede observarse la salida de éste con un osciloscopio. Los osciloscopios del laboratorio permiten, además de realizar la FFT, guardar los resultados en

una memoria externa y en multitud de formatos, de manera que puedan ser extraídos y comparados entre sí.

A tal fin, se ha creado un script en MATLAB que muestra una comparación entre los espectros de las señales resultado de la conversión, permitiendo la comparación y evaluación de opciones. Por otro lado, la forma de las señales en el tiempo ha permitido descartar ciertas opciones por sus malas características, según se verá a continuación.

3.3.4.1 Elección de amplificador operacional y configuración

En primer lugar, se prueba con la placa de conversión en que está soldado el AD9754ARZ, conectándose su salida a una configuración de amplificación inversora con el CLC1005, por lo que se necesita una alimentación externa de -5V. La salida no se vuelve a invertir, de forma que sale negativa.

Los resultados obtenidos son muy malos, según se observa en la Figura 16, donde se usaba un generador de señales triangulares de 50 kHz.

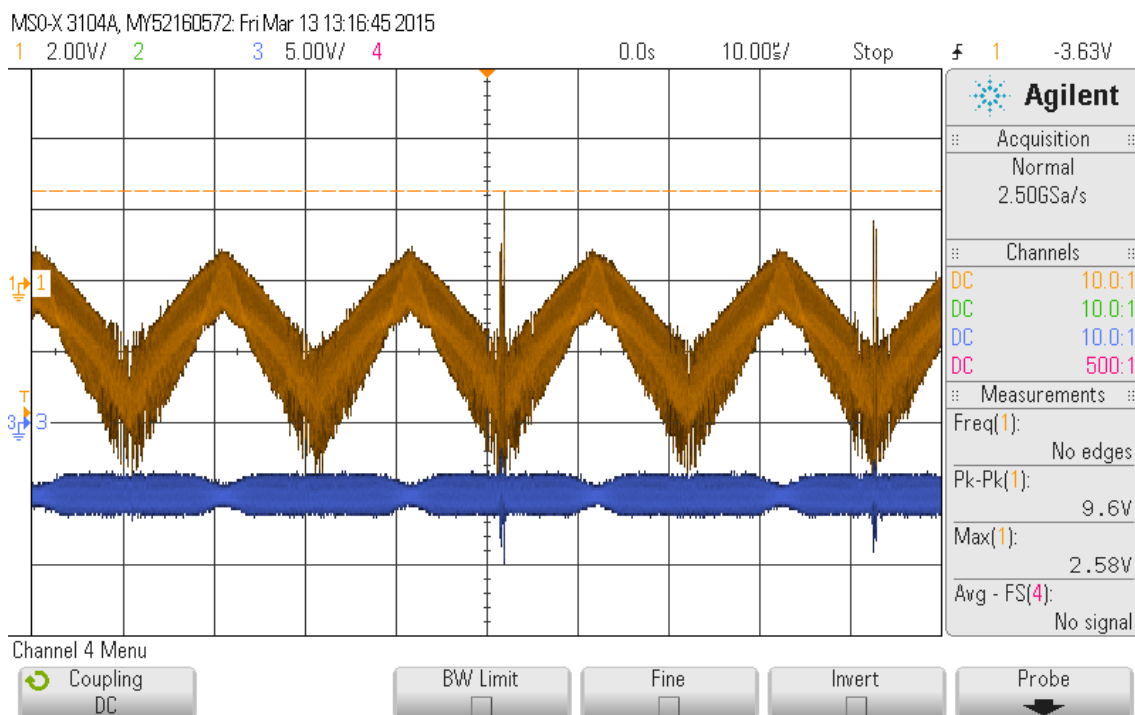


Figura 16: Ruido en configuración inversora

La señal azul representa el ruido que aparece entre la referencia de tierra del circuito diseñado, y la de la FPGA de donde se obtiene.

Se prueba entonces el funcionamiento en continua, con el generador de señales de tipo escalón, y se observa, al ampliar la señal obtenida, que sigue apareciendo un ruido tan grande que deforma totalmente la señal. El ruido aparece incluso cuando la señal debería ser nula, y fuerza una forma de onda aproximadamente periódica y de período igual al de la señal de reloj utilizada (50 MHz).

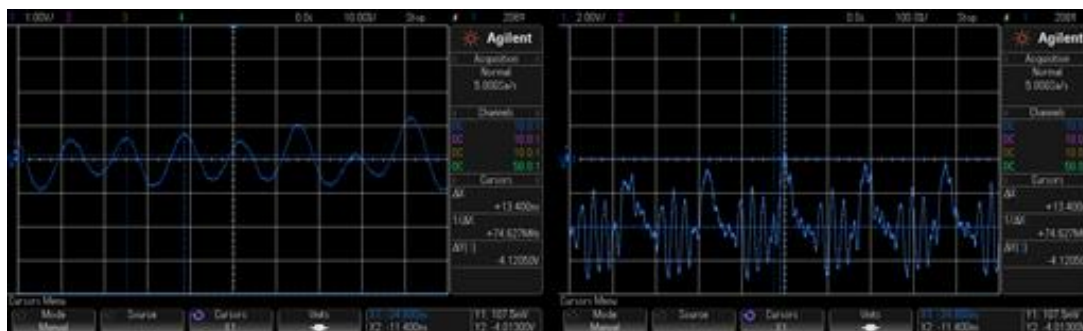


Figura 17: Ruido en una señal continua. Izquierda, todos los bits a 0. Derecha, los 8 bits más significativos a 1

Sustituir la placa en la que está soldado el AD9754ARZ por otra en la que está el DAC904U da el mismo problema, y lo mismo ocurre cuando se cambia el operacional. Se descarta en consecuencia la configuración de amplificador inversor.

Se prueba entonces una configuración no inversora. Cuando se utiliza un seno de 50 kHz de frecuencia, el comportamiento del operacional ADA4857 resulta ser mucho peor que el del CLC1005 (ver Figura 18).

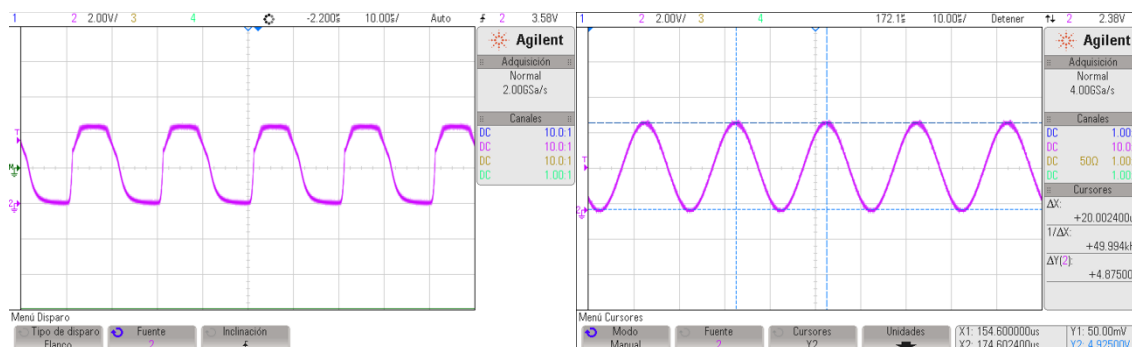


Figura 18: Seno a 50 kHz. Izquierda: ADA4857, derecha CLC1005. DAC THS5651AIDW

Ante la posibilidad de que el problema se deba a una mala soldadura en la placa que contiene al amplificador ADA4857, éste se desuelda y se sustituye por un CLC1005, y el problema desaparece (ver Figura 19). Se pone de manifiesto en consecuencia que el problema es debido al operacional ADA4857, que se descarta. Queda determinado así el uso de la configuración no inversora y del operacional CLC1005.

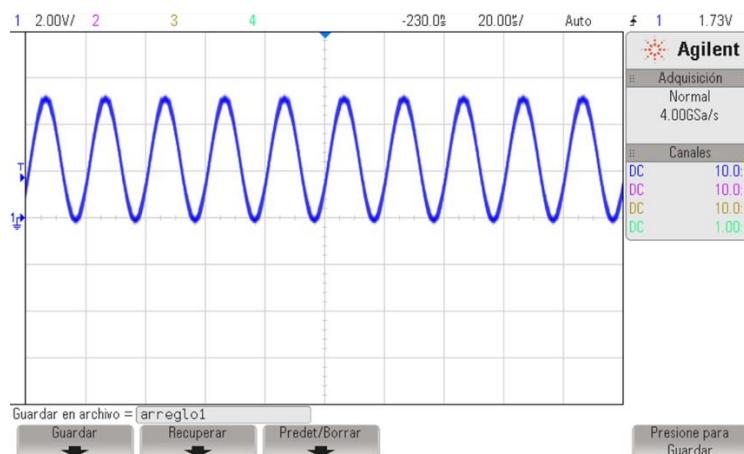


Figura 19: Resultado de sustituir el ADA4857 por el CLC1005

3.3.4.2 Elección de conversor digital-analógico

Para la comparación entre DACs, como ningún caso ofrece un resultado tan negativo como los mostrados en el apartado anterior, se hace necesario un análisis más detallado de la señal de salida, incluyendo su espectro. En consecuencia, se utiliza la opción del osciloscopio de generar la FFT de la señal que se mide, tomándose datos sobre el espectro tanto en banda estrecha como en banda ancha para todos los conversores y señales generadas (a excepción de la señal de continua). Se utiliza una ventana de Hann (o enventanado de *Hanning*), que ofrece buenas características con respecto a la distorsión del espectro de la señal.

La banda estrecha para las señales de 50 kHz comprende el rango desde 0 hasta 2 MHz, de forma que puedan verse los armónicos del tono principal. Para las señales de 1 MHz comprende hasta 20 MHz, con el mismo propósito. Por otro lado, la banda ancha comprende hasta 1 GHz tanto para las señales de alta como de baja frecuencia, con el objetivo de observar el ruido procedente de la frecuencia de reloj (50 MHz o 100 MHz según corresponda a señales de baja o alta frecuencia respectivamente), y sus armónicos.

Se introducen los datos obtenidos en el osciloscopio en un script MATLAB para mostrarlos conjuntamente, y se comparan para elegir el más adecuado a la aplicación. Las gráficas comparativas de espectro y señales temporales pueden consultarse en detalle en el ANEXO A.

El modelo de DAC que peores características parece ofrecer es el THS5651AIDW, en el que tanto en las señales temporales como en el espectro se observa que la señal se ha degradado en gran medida, pues hay muestras que se pierden en la señal, y muchas frecuencias espurias de elevada amplitud, incluso en señales que por sus características deberían ser representadas de forma fiel (ver Figura 20).

Por su parte, el AD9754ARZ ofrece mejores prestaciones que el anterior, pero peores que los otros dos, pues su espectro en banda ancha contiene muchas más frecuencias parásitas, lo que significa un deterioro de la forma de la señal en el tiempo, sobre todo en las señales que no son un tono. Adicionalmente, la corriente de escala completa que genera es menor que el valor teórico, lo que implica que la tensión de la señal de salida es menor que lo esperado (y menor que en el resto de casos). Finalmente, el número de muestras que pierde también es mayor que en el AD9752ARZ y el DAC904U. En la Figura 21 es sencillo ver este problema, además de las peores prestaciones del THS5651AIDW.

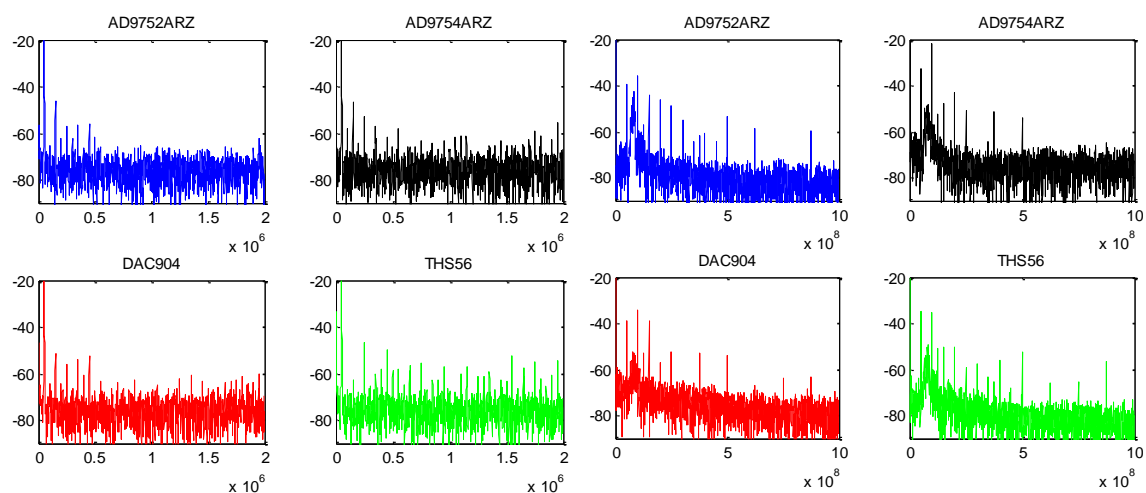


Figura 20: Espectro de una senoide de 50 kHz. Izquierda: banda estrecha. Derecha: banda ancha

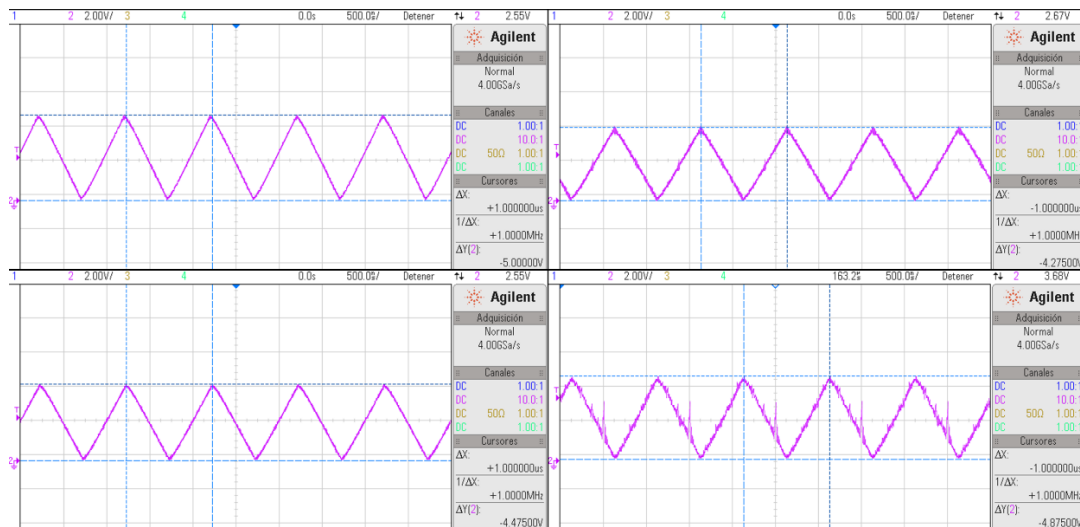


Figura 21: Señal triangular de 1 MHz en el tiempo. Arriba izquierda: DAC904U. Arriba derecha: AD9754ARZ. Abajo izquierda: AD9752ARZ. Abajo derecha: THS5651AIDW

La decisión entre el DAC904U y el AD9752ARZ es mucho más difícil, pues ambos ofrecen prestaciones muy similares: señales con poco rizado en el tiempo, pocas muestras perdidas y espectros con menores frecuencias espurias que los demás DACs. Finalmente se opta por el DAC904U, pues se considera que su mayor resolución puede ser de utilidad en ciertos casos. Asimismo, la corriente de escala completa que ofrece resulta ser ligeramente superior, por lo que el rango dinámico de la tensión de salida es algo más amplio.

3.4. Diseño y evaluación de placas conjuntas

A través del proceso que se ha explicado en los epígrafes 3.2 y 3.3 se ha llegado a la selección de los componentes y configuración que mejores prestaciones ofrecen para la aplicación en cuestión. Para permitir un diseño más compacto, se desea ahora llevar a la misma placa física los módulos de conversión y de amplificación. Para ello, nuevamente se usará la herramienta *Altium Designer*, de forma que pueda desarrollarse el esquemático, optimizar la colocación o *layout*, y obtenerse los archivos que permitan la construcción del PCB. Pueden consultarse los esquemáticos y *layouts* de las placas en el ANEXO A.

Adicionalmente, se presenta la posibilidad de diseñar la placa en desarrollo de forma que sea compatible con la placa de prototipado SpCard de la empresa SpControl. En la práctica, esto se traduce en colocar un conector compatible con los de la SpCard, y un divisor resistivo a la salida, cuestiones que se desarrollan más en detalle a continuación (apartado 3.4.1).

Una vez construida la placa, será sometida a las mismas pruebas que se usaron para los circuitos separados (ver apartado 3.3.3), y finalmente se integrará con el modelo del convertor desarrollado en el CAPÍTULO 4, para observar el funcionamiento del modelo HIL completo.

3.4.1. Compatibilidad con SpCard

Como se ha adelantado, uno de los requisitos que debe cumplir la placa conjunta a diseñar es que pueda conectarse a uno de los conectores de salida de la SpCard, de forma que puedan realizarse modelos HIL usando la FPGA Zynq de la placa como soporte para sintetizar el modelo digitalizado de la planta. Si el funcionamiento del sistema es apropiado, un proceso de mejora de la placa de conversión podría llevar a su integración dentro de la SpCard con su correspondiente explotación comercial.

La placa SpCard permite sacar sus señales digitales al exterior a través de dos conectores macho tipo IDC de 24 pines (CN1 y CN7), cuyo *pinout* se muestra en la Figura 22. Permite además la entrada de señales analógicas a través de otro conector como los anteriores (CN2: ver Figura 22). Las señales analógicas se llevan directamente a un convertor analógico-digital embebido en la placa.

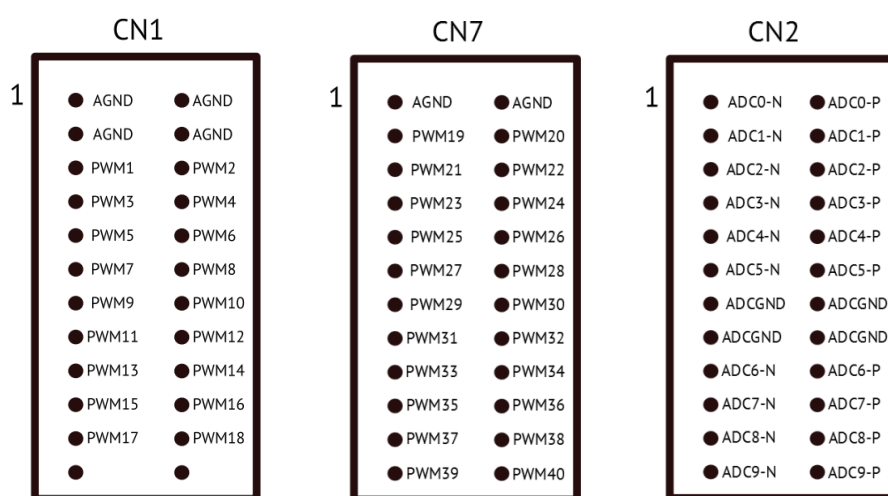


Figura 22: Pinout de conectores de la SpCard

El conector elegido para la placa que se diseña deberá ser por tanto compatible con los mostrados, y la asignación de pines deberá permitir que el sistema funcione tanto si la placa se conecta al conector CN1 como si se conecta al CN7. Se decide finalmente usar otro conector IDC macho de 24 pines para la placa, y permitir la conexión con la SpCard a través de dos conectores hembra unidos con cinta plana. El *pinout* del conector de la placa diseñada se muestra en la Figura 23. Con el objetivo de permitir alimentaciones externas, se incluyen dos pines adicionales separados del conector. Dos tiras de tres pines permiten elegir, colocando un *jumper* entre ellos, si la alimentación de la placa proviene de los pines del conector o de los pines externos (Figura 24)

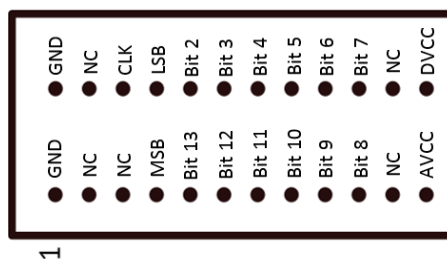


Figura 23: Pinout del conector

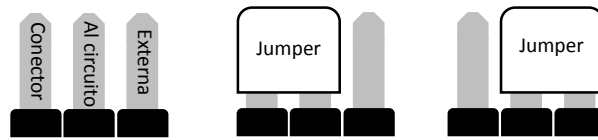


Figura 24: Selección de alimentación. Centro: desde el conector. Derecha: externa

Debido al enfoque de plataforma de desarrollo para reguladores digitales de la SpCard, no existen salidas analógicas de salida, y las señales digitales de salida son PWMs, de forma que toda salida digital de la FPGA se lleva a un comparador. Como se desea conectar la placa de conversión diseñada a la SpCard, es necesario permitir que las señales digitales lleguen al conector saltándose el comparador, o bien hacer que la señal con la que se comparan sea constante y de valor igual al umbral de separación entre la tensión correspondiente a un 1 lógico y la correspondiente a un 0 lógico. Esta segunda opción, aunque permite que no sea necesario modificar físicamente la SpCard, tiene la desventaja de que los comparadores son lentos, de forma que una señal digital de alta frecuencia se verá distorsionada.

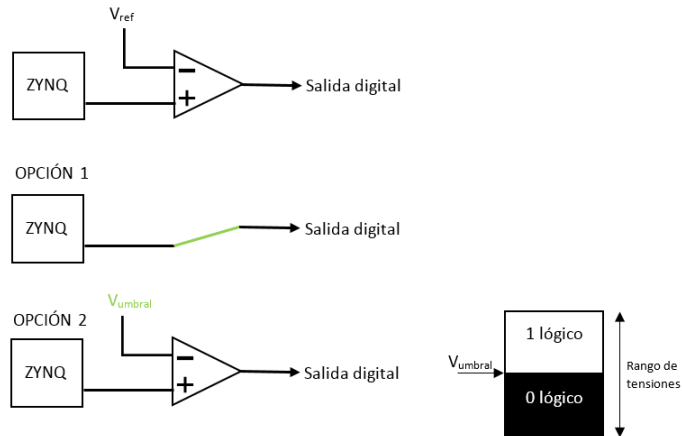


Figura 25: Modificaciones en salidas digitales para permitir compatibilidad

En cualquier caso, estas modificaciones deberían realizarse sobre la SpCard, lo que queda fuera del propósito del presente trabajo.

Una problemática adicional es la que plantea el convertor ADC de que dispone dicha placa, pues sólo acepta señales de hasta 1 V. Debido a que el circuito de conversión-amplificación en diseño ofrece tensiones de 0 a 5 V a su salida, se hace imprescindible incluir una salida adicional en que las tensiones sólo varíen en el rango aceptado por el ADC. Con este propósito, se añade al circuito un divisor resistivo capaz de dividir por 5 la salida. Para ello, se colocan en configuración de divisor de tensión dos resistencias. Los valores comerciales que permiten la relación adecuada son 680 Ω y 2k7 Ω , de forma que las salidas de la placa quedan:

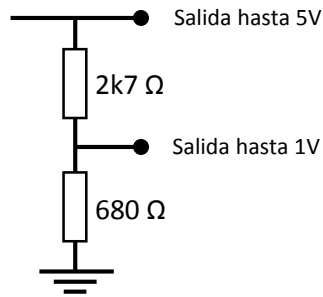


Figura 26: Configuración de salida

Determinados los cambios que deben aplicarse a la placa conjunta, se comienza su proceso de diseño.

3.4.2. Uniformización de pistas

El primer paso en el diseño de la placa conjunta es sencillamente unir los esquemáticos de los circuitos de conversión y amplificación. Añadir los cambios mencionados anteriormente únicamente requiere modificar los conectores e incluir el divisor resistivo de la salida, además de las tiras de pines que permiten elegir entre la alimentación del conector o externa.

El hecho de que existan más componentes demandando corriente en la misma placa hace que se decida incluir también dos condensadores de desacoplo adicionales de capacidad elevada ($1\mu\text{F}$), para tratar de reducir las posibles inductancias parásitas de la red de alimentación a bajas frecuencias. Dichos condensadores, uno para la alimentación analógica y otro para la digital, se colocan lo más cerca posible del punto de entrada de alimentación (en la práctica, del pin central de las tiras de pines de selección de alimentación). Los condensadores de desacoplo de 100 nF que ya se habían utilizado en las placas separadas se colocan lo más cerca posible de los componentes principales del circuito: DAC y amplificador, y favorecerán el buen funcionamiento de la red de desacoplo a altas frecuencias.

Con el objetivo de reducir al máximo el *skew* que aparece entre las señales digitales de entrada a la placa y el reloj, se intenta en un primer diseño uniformizar la longitud de las pistas que llevan esas señales desde el conector al conversor digital-analógico. Para ello, se realizan giros innecesarios en las pistas, que pueden llevar al aumento de capacidades parásitas y por tanto a ruidos, lo que motiva un diseño posterior sin uniformización de pistas. Puede verse en la Figura 27 (izquierda) los giros introducidos en las pistas con el objetivo mencionado. El *layout* completo puede verse en el ANEXO B.

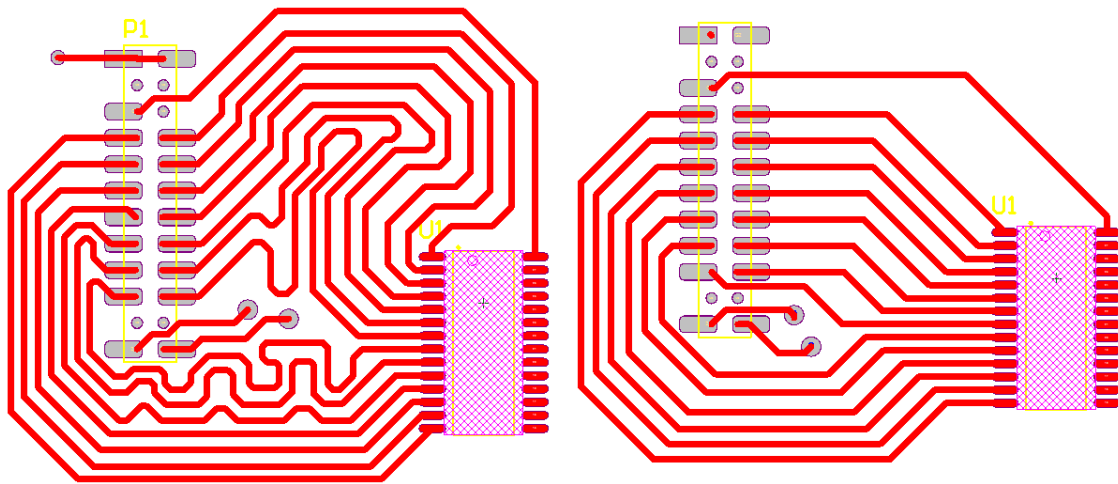


Figura 27: Uniformización de longitud de las pistas. Izquierda: diseño con uniformización. Derecha: sin uniformización

Según se ha adelantado, con el objeto de evaluar si tiene más influencia en la degradación de la señal el *skew* entre señales y reloj, o el aumento de capacidades y resistencias parásitas debido al proceso de uniformización de longitudes, se realiza un nuevo rutado que busca minimizar la longitud de las pistas que llevan las señales digitales desde el conector al conversor. El esquemático de ambos diseños es, por lo demás, idéntico. En la Figura 27 se observa la modificación del *layout*. Ambos diseños se construyen y se prueban, como se describe en el apartado siguiente.

3.4.3. Pruebas y resultados de placas conjuntas

Con ambos diseños construidos y los componentes soldados, se conectan a la placa Spartan 3 Starter Kit Board, en la que se sintetizan los generadores de señales explicados en el punto 3.3.3.1. Se llevan a cabo las medidas de espectro en ambos casos, y se desarrolla un script MATLAB para comparar visualmente los resultados, de forma análoga a como se hizo para las placas separadas. Aunque los resultados completos pueden consultarse en el ANEXO A, se muestra un ejemplo en la Figura 28.

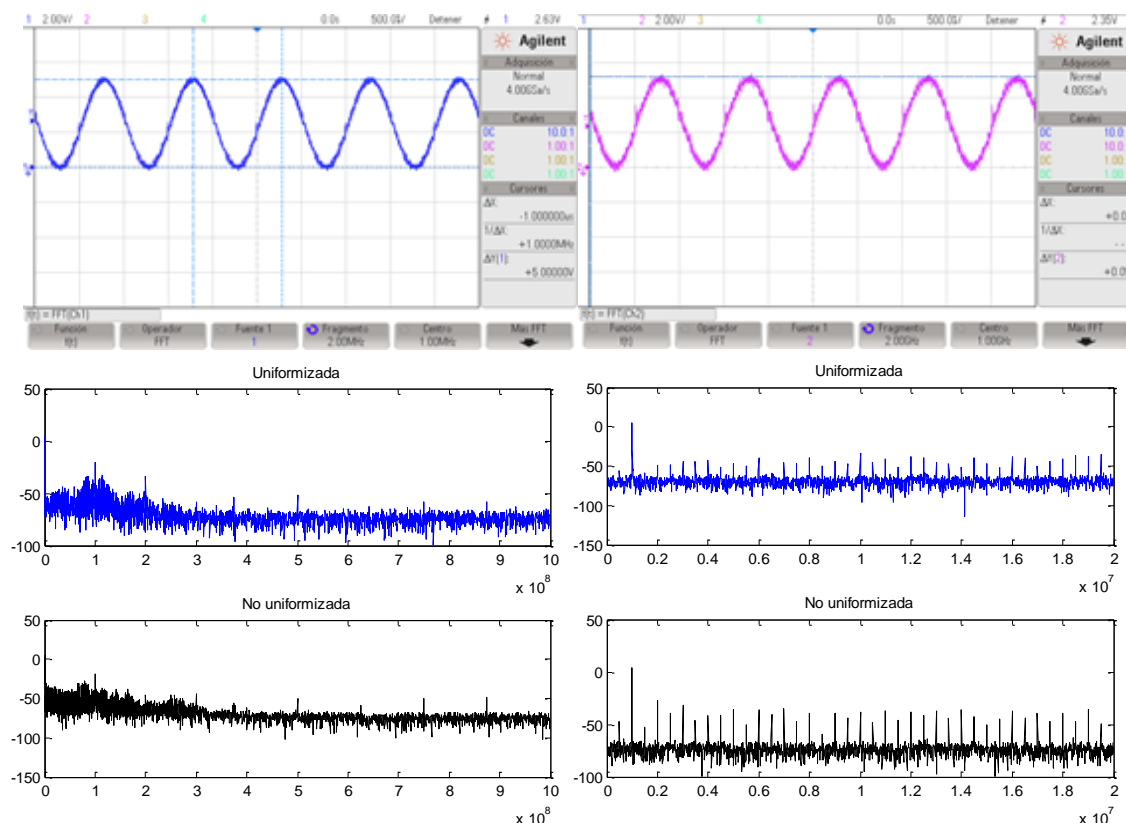


Figura 28: Seno de 1 MHz: Arriba izquierda con uniformización de pistas, arriba derecha sin uniformización. Abajo izquierda: espectro de banda ancha. Abajo derecha: espectro en banda estrecha

Lo primero que se observa al ver la imagen es que la integridad de la señal de salida es peor que la que se obtenía cuando las placas de conversión y amplificación estaban físicamente separadas. Se considera que hay dos factores que pueden explicar esta degradación de rendimiento, que es precisamente la que se ha tratado de evitar con el proceso de diseño que se ha seguido.

En primer lugar, utilizar un conector que busca ser compatible con la SpCard según se ha explicado en el punto 3.4.1 implica que la conexión con la placa de la FPGA Spartan 3 se hace a través de pistas mucho más largas (a la longitud de los cables que unen el conector con la placa de la FPGA se añade la de la cinta plana, y la de las propias pistas del PCB), pudiendo aparecer *crosstalk*, maximizarse el efecto del *skew* entre pistas, o producirse otras perturbaciones electromagnéticas. Las diferencias de montaje pueden apreciarse en la Figura 29.

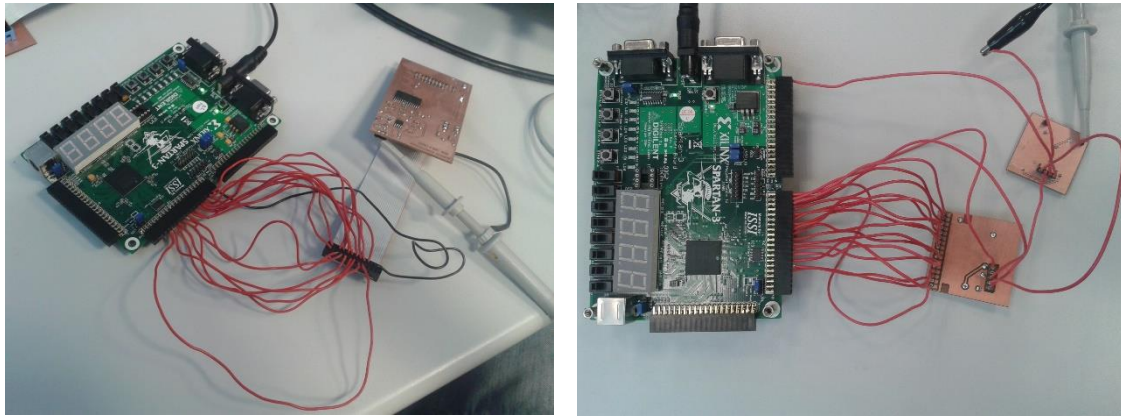


Figura 29: Montaje del sistema. Izquierda: placa conjunta. Derecha: placas separadas

Paralelamente, el hecho de que los planos de tierra de ambos módulos (conversión y amplificación) estén unidos en la placa conjunta puede tener un papel protagonista en la degradación de la señal. La parte digital de la etapa de conversión, con su alta velocidad de conmutación, puede introducir perturbaciones importantes en la parte analógica de amplificación, muy sensible al ruido digital, con la que comparte plano de masa. Si ambos planos están físicamente separados, habiendo una única vía de comunicación, como ocurre cuando las etapas están separadas en placas diferentes, el efecto disminuye y el ruido que aparece es menor. Un efecto muy similar ocurre con la alimentación analógica, con el agravante de que en lugar de ser distribuida por un plano completo, es una pista la que la distribuye por todo el circuito, siendo por tanto muy susceptible a las inductancias parásitas del circuito y al *Vcc-sag*.

Independientemente de esta diferencia con respecto a las placas separadas, la Figura 28 permite apreciar que en el caso en cuestión (seno de 1 MHz), el diseño con uniformización de longitud de pistas ofrece unas prestaciones ligeramente superiores. Se aprecia en la señal temporal que cuando la longitud de pistas no es uniforme, hay ciertos momentos en los que el bit que tiene que cambiar tarda algo más de lo que debe, lo que se traduce en un pico en la señal de salida más acusado cuanto más significativo es el bit que se ha retrasado. Si la longitud de todas las pistas es más parecida, el efecto se reduce, pues lo que se busca no es que las señales tarden más o menos, sino que todas tarden aproximadamente lo mismo.

Para evitar que estos efectos puedan deberse a diferencias en la longitud del cableado con el que el conector se une a la FPGA, se usa en ambos casos exactamente el mismo cable para cada señal. En cualquier caso, no se trata de una diferencia demasiado acusada, máxime si se observan los resultados del resto de señales (ver ANEXO A).

CAPÍTULO 4: Diseño del modelo digitalizado de un convertidor Buck

Según se comentó en el CAPÍTULO 2, el otro elemento de un modelo HIL de cualquier planta analógica es su modelo digitalizado. En este caso, se va a desarrollar el modelo de un convertidor DC-DC reductor de tipo *Buck*, para lo cual inicialmente se determinarán las ecuaciones que rigen su funcionamiento para poder discretizarlas [Jurado] (apartado 4.1). Aunque los primeros diseños serán los de un circuito unifase (apartado 4.2.1), una vez se haya comprobado el buen funcionamiento del sistema a través de su comparación con un modelo analógico de la planta simulado en Simulink, se pasará a un diseño cuadrifase, que ofrece ciertas ventajas con respecto al rizado de la corriente de salida (ver apartado 4.2.2).

Para facilitar las primeras etapas de diseño, el formato de las señales usadas será REAL. Cuando se haya comprobado el correcto funcionamiento del modelo de 4 fases, se realizará un estudio de resolución encaminado a que las señales usadas estén en coma fija, necesarias para conseguir funcionamiento a alta velocidad, y que el modelo pueda ser sintetizado (apartado 4.4).

Finalmente, se describen las modificaciones que ha sido necesario realizar sobre el diseño para permitir su síntesis y correcto funcionamiento sobre la FPGA utilizada (FPGA Spartan 3 en la placa Starter Kit Board), de cara a la integración final del modelo HIL descrita en el CAPÍTULO 5.

4.1. Convertidor Buck, ecuaciones y discretización

Un convertidor Buck es un tipo de regulador de tensión conmutado capaz de transformar un valor de tensión variable a su entrada, en uno más bajo y constante a su salida. Su circuito es el mostrado en la Figura 30, con los sentidos de corrientes indicados.

Su funcionamiento se basa en la capacidad de condensadores y bobinas de almacenar energía sin disiparla, de forma que su rendimiento (la relación entre la energía que entra al circuito y la que sale) es del 100% si no se consideran las pérdidas.

Un análisis detallado del funcionamiento permite obtener las ecuaciones que determinan su comportamiento.

En primer lugar, es necesario explicar que los MOSFETs no tienen un comportamiento independiente. Cuando el MOSFET superior (en adelante HSM por sus siglas en inglés *High Side Mosfet*) está en estado de conducción, el inferior (o LSM, de *Low Side Mosfet*) debe estar en estado de no conducción, o se produciría un cortocircuito. Si es el LSM el que conduce, el HSM no deberá conducir, por el mismo motivo. Se da un caso especial en determinados momentos normalmente llamados *tiempos muertos* en que ninguno de los transistores conduce. Estos tiempos muertos aparecen porque los transistores tardan unos instantes en cambiar de estado de conducción a no conducción y viceversa. En esos instantes, el comportamiento del circuito lo determinan los diodos colocados en paralelo con los condensadores.

Con las pautas indicadas arriba, puede analizarse el comportamiento del circuito supuesto sin pérdidas. Se distinguen tres posibles casos:

A) HSM conduce, LSM no conduce

Se cumple entonces que la corriente de entrada es la que atraviesa la bobina, y la tensión que cae en la bobina es igual a la tensión de entrada menos la de salida:

$$I_{in} = I_L \text{ y además } V_L = V_{in} - V_{out} = L * \frac{dI_L}{dt}, \text{ por tanto } V_{in} - V_{out} = L * \frac{dI_{in}}{dt}$$

Si queremos discretizar la ecuación diferencial para convertirla en una ecuación en diferencias, tendríamos:

$$V_L = V_{in} - V_{out} = L * \frac{\Delta I_{in}}{\Delta t} \quad (i)$$

Siendo Δt el *paso de integración*, o el inverso de la *frecuencia de cálculo* de valores.

B) HSM no conduce, LSM conduce

En este caso, tenemos que $I_{in} = 0$ y que $V_L = -V_{out} = L * \frac{dI_{in}}{dt}$ por tanto la ecuación discretizada queda:

$$V_L = -V_{out} = L * \frac{\Delta I_{in}}{\Delta t} \quad (ii)$$

En ambos casos, $I_C = I_L - I_{out} = C * \frac{dV_{out}}{dt}$ de donde, discretizando y despejando,

$$\Delta V_{out} = (I_L - I_{out}) * \frac{\Delta t}{C} \quad (iii)$$

C) Ninguno conduce: tiempo muerto

El comportamiento del circuito queda determinado por los diodos colocados en antiparalelo con los MOSFET, y por la corriente que recorría la bobina cuando se entró en el tiempo muerto. Si la corriente que recorría la bobina era positiva, el diodo en antiparalelo con LSM conducirá pero el de HSM no, de forma que el caso será equivalente al B). Si la corriente es negativa, ocurre lo contrario: el diodo en antiparalelo al HSM conduce pero no lo hace el que está en paralelo con LSM: el comportamiento es el del caso A).

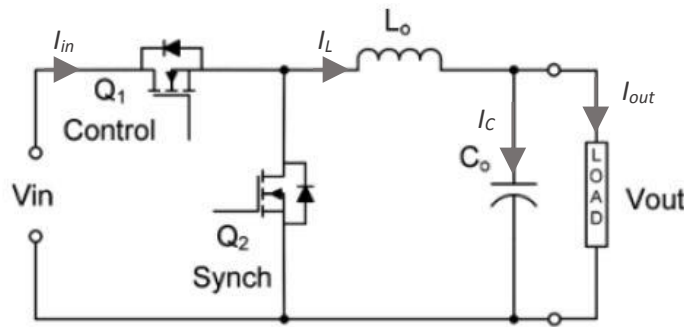


Figura 30: Convertidor Buck unifase

Con las ecuaciones descritas, ya se puede modelar el funcionamiento del convertidor.

En primer lugar, un análisis en régimen permanente determina que $\Delta I_{inA}) = -\Delta I_{inB})$ o en caso contrario, la corriente crecería en magnitud de forma indefinida. En consecuencia, igualando (i) y – (ii):

$$(V_{in} - V_{out}) * \Delta t_A = V_{out} * \Delta t_B \Rightarrow V_{out} = \frac{V_{in} * \Delta t_A}{\Delta t_A + \Delta t_B}$$

Generalmente, a Δt_A se le da el nombre de T_{on} porque se refiere a una configuración alternativa en la que el MOSFET inferior es sustituido por un diodo, de forma que los períodos pueden nombrarse en función de si el único MOSFET restante está conduciendo (ON) o no conduciendo (OFF).

Por otro lado, ya que en definitiva todo el tiempo o se está en el estado A) o B), $\Delta t_A + \Delta t_B = T_{total}$, es decir, el ciclo de conmutación completo. Al cociente T_{on} / T_{total} se le denomina *ciclo de trabajo*, se le designa por la letra D (de *duty cycle*, su nombre en inglés), y representa la fracción total de tiempo que está conduciendo HSM sobre el tiempo que constituye un período completo de conmutación.

La ecuación anterior queda entonces

$$V_{out} = V_{in} * D \quad (iv)$$

Que es la ecuación fundamental que permite comprender el funcionamiento del convertidor: la tensión de salida, en valor medio, es igual a la de entrada multiplicada por el ciclo de trabajo. Cualquier regulador se encargará de modificar el ciclo de trabajo para mantener la salida constante en el valor requerido.

La corriente que recibe la carga es igual al valor medio de la que atraviesa la bobina, como puede observarse haciendo un análisis en continua del circuito. Esta última tiene un rizado mayor cuanto mayor es el período de conmutación, que se traduce en rizado en las corrientes de entrada y de salida. Esto implica que la tensión y la corriente de entrada no sean proporcionales, y la tensión y corriente de salida tampoco, cuestiones de importancia al aplicar corrección del factor de potencia. Un buen diseño multifase permite reducir el rizado de la corriente suma de cada fase reduciendo el problema, según se comentará en 4.2.2.

Las ecuaciones discretizadas (i), (ii) y (iii) son las que se implementan en VHDL, usando (iv) para saber cuál debe ser el ciclo de trabajo para la tensión de salida deseada. Una vez definido el valor de la tensión de entrada y de la carga, que determina el valor de la corriente de salida, se tiene completamente caracterizado el comportamiento del circuito.

4.2. Modelo REAL con reloj de 50 MHz

Para poder comenzar con el desarrollo del modelo en VHDL sin necesidad de realizar un estudio de resolución, se comienza usando señales en formato REAL. Este formato de representación sigue el estándar de señales de coma flotante de doble precisión (64 bits), pero no es sintetizable sobre una FPGA. Se usa únicamente para comprobar que se ha elaborado correctamente el modelo, y posteriormente para evaluar la pérdida de precisión que representa la coma fija y sus efectos en la señal.

4.2.1. Diseño unifase

Se utiliza inicialmente un diseño de una única fase, basado en el circuito de la Figura 30. Para ello, se divide el funcionamiento del circuito en dos bloques. En primer lugar, se parte de los posibles casos de funcionamiento del convertidor explicados en el apartado 4.1, y se determinan las ecuaciones que aplican en cada caso, según se muestra en Código 1. Se

selecciona así el valor de la tensión de la bobina y de la corriente de entrada al circuito. Una señal auxiliar de error permite determinar fácilmente en las simulaciones si se ha producido algún estado erróneo. Las señales usadas son en muchos casos señales auxiliares que permiten leer y escribir, y son luego asignadas a los puertos del circuito, que sólo pueden ser escritos o leídos.

```
MULTIPLEXOR: process(HSM, LSM, Ibob, Vout_INTERNA, Iout)
begin
  if HSM='1' and LSM='0' then --Comportamiento de carga
    Vbob<=Vin-Vout_INTERNA;
    Iin_INTERNA<=Ibob;
  elsif HSM='0' and LSM='1' then --Comportamiento de descarga
    Vbob<=-Vout_INTERNA;
    Iin_INTERNA<=0.0;
  elsif HSM='0' and LSM='0' then --Comportamiento residual
    --Entran en juego los diodos acoplados a los mosfet
    if Ibob<0.0 then --Corriente de la bobina negativa: diodo HSM polarizado en directa
      Iin_INTERNA<=Ibob;
      Vbob<=Vin-Vout_INTERNA;
    else --Corriente de la bobina positiva: diodo LSM polarizado en directa
      Iin_INTERNA<=0.0;
      Vbob<=-Vout_INTERNA;
    end if;
  else --Caso imposible, ambos MosFet activos: cortocircuito
    SENALERROR<='1';
  end if;
end process MULTIPLEXOR;
```

Código 1: Selección de modo de funcionamiento

Cuando se han determinado los valores dependientes del modo de funcionamiento, se pueden expresar las ecuaciones en diferencias que reflejan el comportamiento del sistema, y son independientes del modo de funcionamiento.

```
ECDIFFS: process(ClkIn, Reset)
begin
  if Reset='1' then
    Vout_INTERNA<=0.0; --Al inicio el condensador esta descargado
    Ibob<=0.0;
  elsif rising_edge(ClkIn) then
    Vout_INTERNA<=Vout_INTERNA+Icond*inct_C;
    Ibob<=Ibob+Vbob*inct_L;
  end if;
end process ECDIFFS;
```

Código 2: Implementación de ecuaciones en diferencias

Con la asignación de las señales _INTERNA a sus correspondientes puertos, y la definición de I_{cond} como $I_{bob} - I_{out}$, el modelo queda completo. Únicamente resta fijar los valores de los componentes y del paso de integración. Basándose en el trabajo de [Usero], se determina un valor óptimo para los componentes de 22 μ H para la bobina del sistema, y de 220 μ F para el condensador. El paso de integración, o período de cálculo, se establece en 20 ns, equivalente a una frecuencia de cálculo de 50 MHz. La frecuencia de cálculo en el modelo la determina el reloj, de forma que la frecuencia de éste también deberá ser de 50 MHz.

Para poder simular el circuito se necesita además indicarle a los MOSFETs modelados con la selección de modo del Código 1 cuándo deben conducir y cuándo no. En este primer diseño, esto se hace en el *testbench* del sistema. Para ello, se crean unos contadores que van de 0 a 249. Por otro lado, se fija un valor de comparación, de forma que cuando el contador los supere una determinada señal digital binaria cambie de valor. En la práctica, se ha implementado un generador PWM con un ciclo de trabajo fijo, que se conecta a los modelos

de MOSFET- La frecuencia del PWM es 250 veces más baja que la del reloj usado, que ya se comentado que era de 50 MHz. En consecuencia, se tiene una frecuencia de conmutación de 200 kHz.

Finalmente, se determina el valor de la corriente de salida del modelo como la tensión de salida V_{out} multiplicada por la conductancia de salida G_{out} .

Se utilizan finalmente:

- El valor 0,4 siemens para la conductancia de salida, que equivale a $2,5 \Omega$ de resistencia de salida.
- El valor 12 V para la tensión de entrada.
- El valor 5/12 para el *duty-cycle*, que sobre una cuenta de 250 da el valor 104.

De los datos anteriores se llega a que la tensión de salida tendrá un valor final de 5V, y la corriente de salida un valor medio de 2 A en régimen permanente.

En el ANEXO A se puede consultar el código VHDL completo del modelo y el *testbench*. En el ANEXO A puede observarse la simulación de su régimen transitorio, y cómo se alcanza el valor final buscado.

4.2.2. Diseño cuadrifase

Un convertidor Buck cuadrifase tiene un esquema como el mostrado en la Figura 31. Cada fase dispone de su propia bobina y par de MOSFETs de forma que puede regularse de forma independiente a las demás, y todas las fases comparten el condensador de salida. La tensión de salida es la misma a la que se obtendría con un diseño unifase, pero en un buen diseño, las corrientes de entrada y de salida tienen mucho menos rizado que en el caso unifase. Esto es debido a que se introduce un desfase en los ciclos PWM que controlan cada una de las fases, de forma que las corrientes de cada una también estén desfasadas. Si el desfase es el adecuado, la suma de las corrientes permite reducir el rizado de la corriente total, lo que se conoce como *interleaving*, mostrado para un caso de dos fases en la Figura 32.

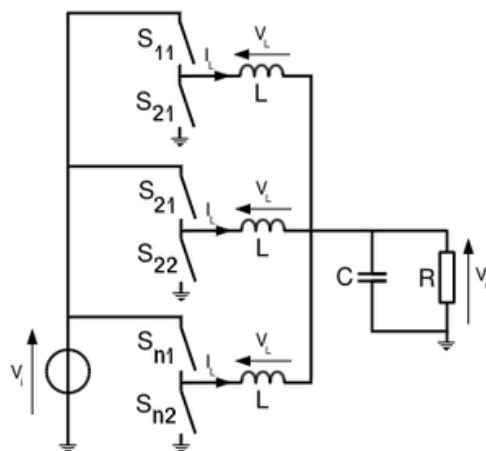


Figura 31: Buck multifase (3 fases)

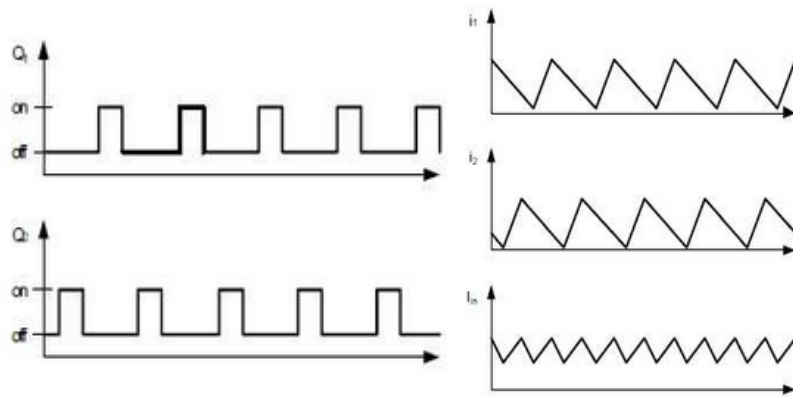


Figura 32: Interleaving en un modelo de dos fases.

Las modificaciones realizadas sobre el modelo anterior pueden resumirse en replicar el código mostrado en Código 1 cuatro veces, una por fase, modificando en cada caso el par de MOSFETs que controla el funcionamiento. A continuación, en el modelado de ecuaciones en diferencias ha de incluirse el resto de fases, como se muestra en el Código 3.

```
ECDIFFS: process(ClkIn, Reset)
begin
  if Reset='1' then
    Vout_INTERNA<=0.0; --Al inicio el condensador esta descargado
    IL1<=0.0;
    IL2<=0.0;
    IL3<=0.0;
    IL4<=0.0;
  elsif rising_edge(ClkIn) then
    Vout_INTERNA<=Vout_INTERNA+Icond*inct_C;
    IL1<=IL1+VL1*inct_L;
    IL2<=IL2+VL2*inct_L;
    IL3<=IL3+VL3*inct_L;
    IL4<=IL4+VL4*inct_L;
  end if;
end process ECDIFFS;
```

Código 3: Modificaciones en las ecuaciones en diferencias para el modelo cuadrifase

Por otro lado, las modificaciones en el *testbench* consisten en generar cuatro PWMs diferentes, uno para cada par de MOSFETs, replicando el código del modelo unifase. Para permitir que el desfase sea el adecuado, el valor de inicio de la cuenta de cada PWM tras un reseteo es diferente: cada PWM empieza un cuarto del valor máximo de cuenta más adelantado que el anterior, de forma que la fase de cada uno cambia en 90 grados, lo que permite que aparezca el mismo desfase en las corrientes de cada fase.

Las simulaciones del régimen transitorio pueden encontrarse en el ANEXO A.

4.3. Simulaciones del circuito analógico

Para poder evaluar el funcionamiento del modelo digital desarrollado, es necesario contar con resultados de una simulación del circuito analógico con los que comparar. Por este motivo, se crea en Simulink el circuito del Buck unifase (ver Figura 33), y se procede a su simulación, que arroja valores muy similares a los que se conseguían con el modelo digital, lo que avala su funcionamiento.

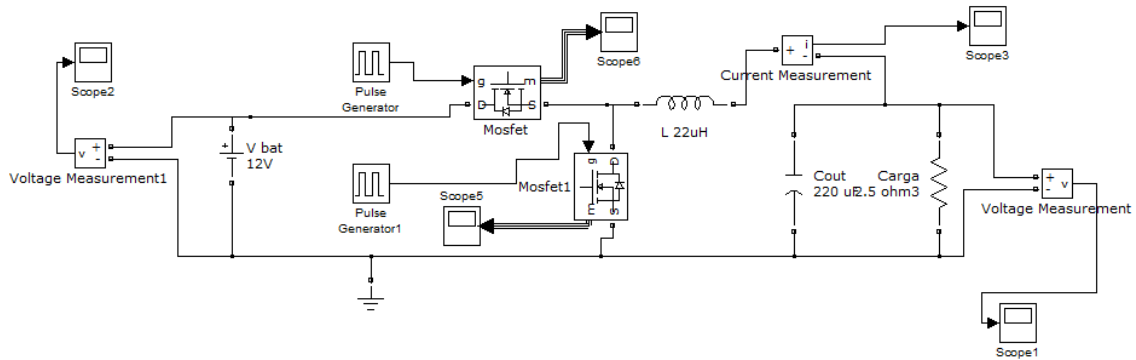


Figura 33: Circuito del Buck en Simulink

Una vez se ha desarrollado el modelo unifase, se pasa a crear un modelo cuadrifase que permita evaluar el modelo digital diseñado. En Simulink, el circuito queda muy similar al unifase, replicando cuatro veces la etapa con los MOSFETs y la bobina. Puede observarse en la Figura 34 el interleaving de corrientes, o cómo la suma de las corrientes desfasadas permite reducir el rizado de la corriente total:

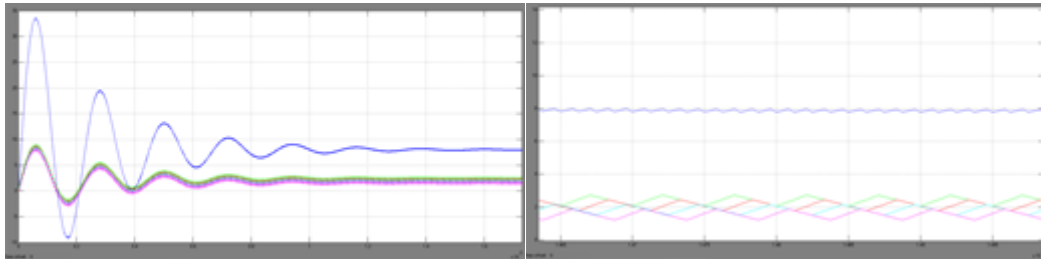


Figura 34: Corrientes del modelo cuadrifase. Azul oscuro: suma de corrientes de las bobinas. Resto de colores: Corrientes de cada fase. Derecha: vista ampliada del régimen permanente.

Nuevamente, los resultados son muy similares a los del modelo REAL. Existen pequeñas diferencias en los valores exactos debido a no idealidades tenidas en cuenta en el modelo analógico (pequeña resistencia parásita de los MOSFET, conductividad no infinita del diodo en paralelo...). Sin embargo, los instantes de tiempo en que se sitúan los picos y valles de sobre-oscilación son idénticos.

4.4. Modelo cuadrifase en coma fija a 50 MHz

Con un modelo del circuito ya discretizado (epígrafe 4.2), y sabiendo que su funcionamiento es correcto gracias a la comparación con una simulación analógica (epígrafe 4.3), puede pasarse a un diseño en el que las señales utilizadas estén en formato de coma fija, de forma que pueda sintetizarse sobre una FPGA.

4.4.1. Biblioteca sfixed y primer modelo en coma fija

Para el desarrollo de este modelo, se utiliza la biblioteca VHDL *sfixed*, que implementa el formato de representación QX.Y con signo, y sus operaciones. Para definir señales de coma fija con esta biblioteca, se utiliza el tipo `sfixed(A downto -B)`, siendo A el número (positivo) de bits destinados a la parte entera sin tener en cuenta el bit más significativo (con información del signo), y B el número (también positivo) de bits dedicados a la parte decimal. Para representar valores muy pequeños en este formato, pueden hacerse declaraciones de tipo `sfixed(-A`

downto -B), en las que se dedican todos los bits a parte decimal, usándose las potencias de dos entre $-A+1$ (el +1 debido al bit de signo) y $-B$. El caso opuesto, `sfixed(A downto B)` permite representar números muy grandes. Un ejemplo de uso se muestra en el Código 4.

```
constant inct_L :      sfixed(-10 downto -27) := to_sfixed(0.000909090,-10,-27);
constant inct_C :      sfixed(-13 downto -30) := to_sfixed(0.000909090,-13,-30);
signal Vout_INTERNA :  sfixed(4 downto -23) := (others => '0');
```

Código 4: Ejemplos de definición de tipo *sfixed*

Para el diseño en coma fija se desarrolla en primer lugar un modelo en el que a las señales se les asigna un número muy elevado de bits, mucho mayor del estrictamente necesario, únicamente para comprobar que el diseño se ha hecho correctamente. En esencia, se sustituyen todas las señales que en el modelo previo eran de tipo REAL por señales *sfixed* con el mismo valor, pero usando un número muy elevado de bits para su representación, por lo que la pérdida de precisión esperada es muy pequeña.

Los puertos de entrada y salida sí que se definen, sin embargo, con un número de bits reducido (16 bits para las corrientes y tensiones de entrada y salida). Su tipo es *std_logic_vector*, de forma que las señales *sfixed* que escriben en los puertos deben cambiar su tamaño a 16 bits, y asignarse mediante la función *to_slv*, mientras que las señales que leen de ellos deben recibir el valor a través de la función *to_sfixed*, ambas funciones definidas en la biblioteca (ver Código 5), al igual que la suma, resta, multiplicación y división de valores en formato *sfixed*. La función *resize*, también de la biblioteca, permite ajustar los valores de señales a un número determinado de bits para la parte entera y decimal, redondeando al valor más cercano, extendiendo el signo o rellenando con ceros de ser necesario.

```
Icond<=resize(SumIL-to_sfixed(Iout,4,-11),Icond);
Iin<= to_slv(resize(Iin_etapa1+Iin_etapa2+Iin_etapa3+Iin_etapa4,5,-10));
Vout<= to_slv(resize(Vout_INTERNA,4,-11));
```

Código 5: Uso de *to_slv*, *to_sfixed* y *resize* para asignar señales a puertos y viceversa

Las señales que se comparan con las del modelo REAL para evaluar la pérdida de precisión son las internas en lugar de los puertos en los que los valores ya se han truncado.

Una vez se dispone de un modelo en coma fija que funciona y arroja resultados muy similares a los del modelo REAL, se realiza un estudio de resolución destinado a reducir el número de bits empleados en la representación de cada señal, para la creación de un modelo en coma fija optimizado. Se describe a continuación el proceso seguido para determinar el número de bits que se usa para representar cada señal o constante.

4.4.2. Estudio de resolución y modelo optimizado

Para las constantes resultado de la discretización de las ecuaciones diferenciales: *inct_L* e *inct_C*, representando $\frac{\Delta t}{L}$ e $\frac{\Delta t}{C}$ respectivamente, se parte del $\log_2 \text{valor}$ para establecer el bit MSB, y el $\log_2 10^{-\text{Posición decimal más pequeña}}$ para saber el bit LSB de la representación en coma fija.

De esta forma, por ejemplo el valor de *inct_L* para los valores de componentes y paso de integración comentados anteriormente es de $9,0909 \cdot 10^{-4}$. El logaritmo en base dos es igual a $-10,103$, por lo que conseguir que el bit más significativo excluyendo el de signo sea la potencia -10 de 2 será suficiente para representar el número (el bit de signo añade el mismo

rango en negativo, no hace que se puedan representar números hasta 2^{-9}). Su posición decimal más pequeña es la que está ocho posiciones por detrás de la coma, de donde $\log_2 10^{-8} = -26.575$. Con la potencia -26 de dos el número más pequeño representable sería $1,49 \cdot 10^{-8}$. Se elige entonces la potencia -27 , de forma que el número más pequeño representable es $0,7451 \cdot 10^{-8}$. Puede establecerse exactamente el error cometido en este caso como el resto de dividir $9,0909 \cdot 10^{-4}$ entre $0,7451 \cdot 10^{-8}$, y es igual a $9,41 \cdot 10^{-10}$, una cifra tan ínfima que puede considerarse despreciable. En cualquier caso, procediendo de esta forma el error siempre está acotado por el menor número representable dividido entre 2, ya que la biblioteca *sfixed* se encarga de redondear al valor posible más próximo.

Para las señales que representan *variables de estado*, es decir, que dependen de valores anteriores de la misma señal de forma que el error cometido se acumula en cada iteración, se observa el valor máximo en el modelo REAL para establecer el MSB de la misma forma que se hacía en las constantes. Para el bit más pequeño, se observa el valor del incremento a sumar en régimen permanente en el modelo REAL, y se calcula el 5% de ese valor. Al resultado se le aplica el logaritmo en base 2, y se le suma 8, obteniéndose así la potencia de 2 que representará el LSB. Se procede de esta manera porque usar el mínimo incremento posible para determinar el número de bits de la parte decimal implicaría una señal con un número de bits demasiado elevado.

Para el resto de señales, se determina en el modelo REAL el valor máximo y se le aplica el logaritmo de base 2 para determinar el MSB, y se calcula $\log_2 10^{-\text{Posición decimal más pequeña}}$ para determinar el LSB, de forma análoga a lo que se hacía para las constantes.

Finalmente, al observar que la FPGA Spartan 3 dispone de multiplicadores embebidos de 18 bits, que permiten acelerar en gran medida los cálculos, se opta por restringir la precisión de algunas señales que intervienen en multiplicaciones a ese número de bits, quedando 17 para la representación y uno para el signo. Las constantes *inct_L* e *inct_C* y las tensiones de las bobinas se representan con ese número de bits, pues las primeras pueden representarse con 18 bits respetando el número de bits que se determinó en un primer momento, y la pérdida de precisión que supone usar 18 bits para representar las tensiones es mínima.

Con las modificaciones comentadas, se simula el modelo para comparar los resultados con los que se obtuvieron tras la simulación del modelo REAL. De esa forma pueden evaluarse los efectos de la pérdida de precisión, que resultan ser muy pequeños (ver Tabla 6, Tabla 7 y Tabla 8).

Se puede observar en la Figura 35 un esquema simplificado del hardware que intervendría al sintetizar el modelo HIL. Se indica además número de bits utilizado para representar cada parte de las señales en coma fija.

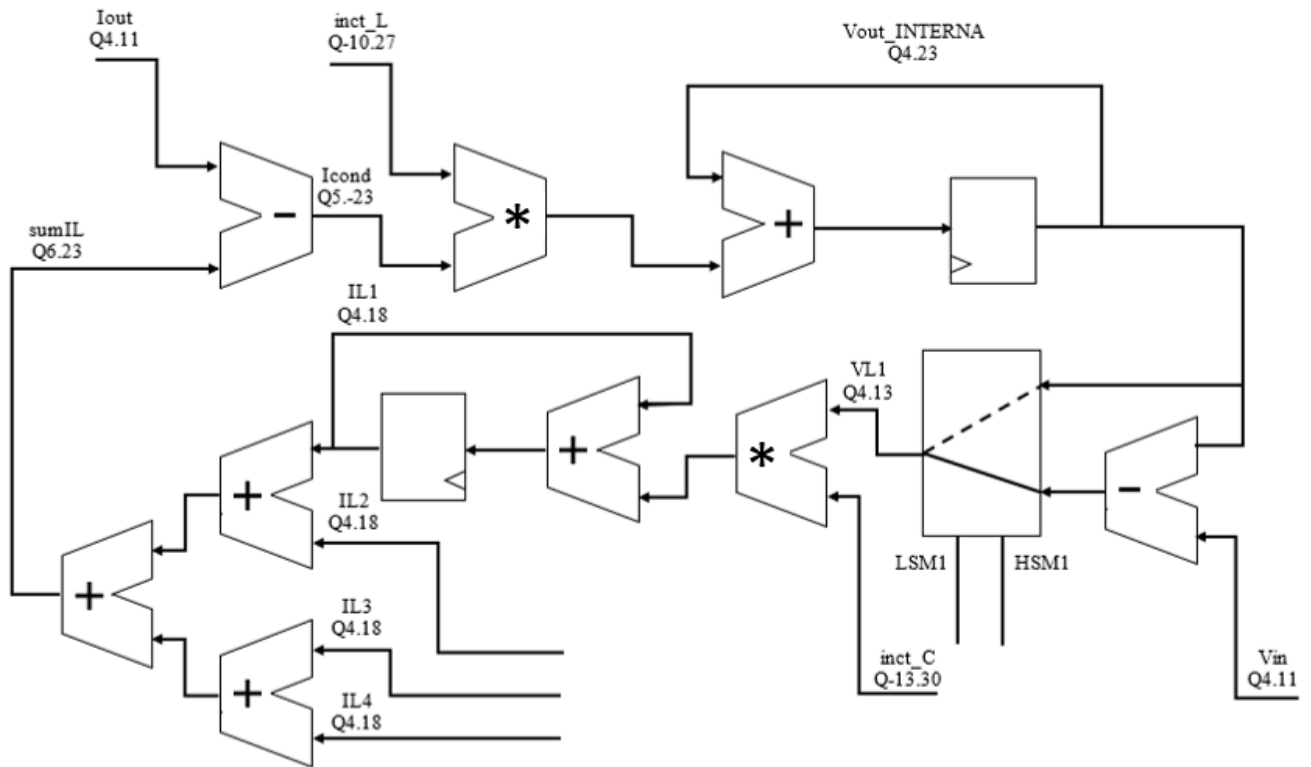


Figura 35: Esquema simplificado del hardware interviniente en el modelo HIL

4.5. Modificaciones para síntesis

Con el modelo ya desarrollado, queda conseguir que pueda ser sintetizado en una FPGA. El modelo de FPGA usado para la síntesis es Spartan 3, colocado en la placa Starter Kit Board. Se planea también su síntesis en un dispositivo Zynq, pero debido a que cuenta con más capacidad y velocidad, si el modelo es sintetizable sobre una Spartan 3, su síntesis en una Zynq no presentará problemas.

En primer lugar, es necesario establecer un *constraint* de reloj, es decir, una condición de tiempo que el circuito debe cumplir: concretamente funcionar adecuadamente con un reloj simétrico de frecuencia 50 MHz. Esta condición se especifica en un fichero UCF (*User Constraints File*), donde además se indica cuál es el mapeo de los puertos de entrada y salida del modelo a pines de entrada y salida de la FPGA (ver Código 6).

```
//Constraint de reloj
NET "CLK" TNM_NET = CLK;
TIMESPEC TS_CLK = PERIOD "CLK" 20 ns HIGH 50%;
```

Código 6: Especificación de la condición de tiempo que debe cumplir el circuito

Paralelamente, para permitir su síntesis, el circuito se separa en tres bloques independientes en ficheros VHDL diferentes, instanciados en un *top level*. El primero es el bloque que modela el comportamiento del Buck, siendo el verdadero modelo digitalizado de la planta. El segundo sirve para modelar la alimentación de entrada (en el caso en cuestión, se deja constante de valor 12 V) y la carga, que se modela como una resistencia de valor 0,625 Ω y un retardo de un ciclo. El tercero se encarga de generar los pulsos PWM desfasados necesarios para el funcionamiento del modelo, trasladando una función que antes realizaba el

testbench a un módulo adicional. Debido a que los ciclos PWM tienen un ciclo de trabajo fijo, que no cambia en función de la salida del modelo digitalizado, el sistema así implementado está funcionando en lazo abierto. El esquema del sistema completo se muestra en la Figura 36.

Aunque se desarrolla un *top level* para que todos los módulos puedan ser implementados e interconectados en la misma FPGA, podrían ser sintetizados cada uno sobre una FPGA diferente, que deberían ser conectadas externamente.

Cuando se intenta sintetizar sobre la Spartan 3 el sistema con las modificaciones que se han mencionado, el software indica que no es posible cumplir los requerimientos de tiempo: el circuito no puede funcionar a 50 MHz si no se realiza alguna modificación, pues sólo el tiempo necesario para realizar operaciones (sumas, redondeos, multiplicaciones, lógica...) ya supera el permitido, incluso sin tener en cuenta el tiempo que tardan las señales en llegar de un punto a otro.

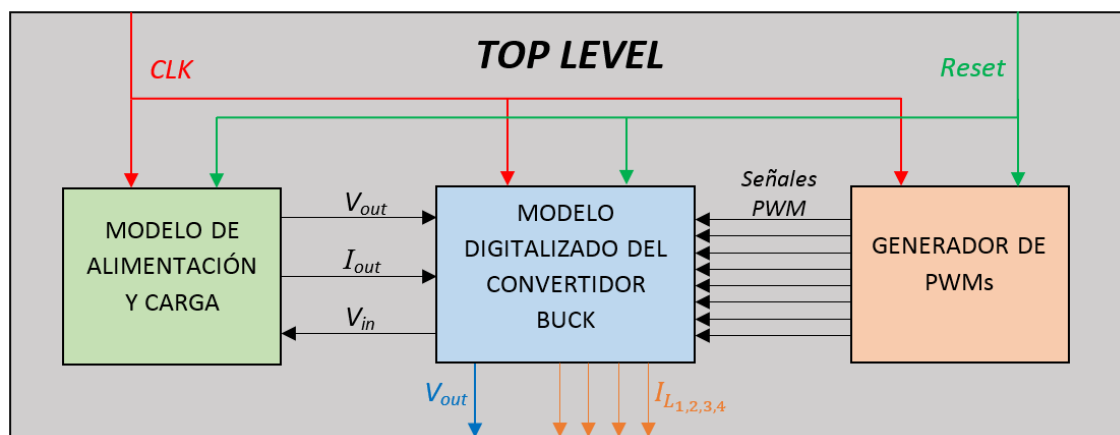


Figura 36: Sistema completo apto para ser sintetizado

4.5.1. Soluciones al problema de cumplimiento de tiempos

Para tratar de solventar el problema se adoptan dos enfoques muy diferentes. El primero consiste en realizar una segmentación o *pipeline* del modelo, manteniendo la frecuencia de cálculo de 50 MHz. Esto significa que el modelo no va a ser igual que una planta analógica en la que no se introducen ciclos de retardo entre componentes, pero si la frecuencia es lo suficientemente elevada –y se considera que 50 MHz lo es –, y el número de ciclos de retardo es pequeño, el error será de poca magnitud y el modelo seguirá siendo válido. Esta posibilidad se explica más en detalle en el epígrafe 4.5.1.1.

La alternativa consiste simplemente en reducir la frecuencia de funcionamiento del circuito, bajando de 50 MHz a 20. Esto supone modificar tanto el modelo digitalizado del Buck como el generador de PWM, debido a que se mantiene el período de conmutación del circuito. Se habla en profundidad de esta solución en el epígrafe 4.5.1.2.

Para evaluar ambas posibilidades, se comparan valores de determinados instantes del régimen transitorio de tensiones y corrientes, y se determina cuál de las soluciones se acerca más a su modelo REAL. Puede verse el proceso en el epígrafe 4.5.1.3.

4.5.1.1 Modelo a 50 MHz con segmentación

Según se ha comentado, la primera solución consiste en introducir una segmentación en el circuito. Segmentar significa introducir etapas de registros entre operaciones lógicas, de forma que entre dos etapas de registros consecutivas las operaciones puedan realizarse en un tiempo menor que el período de reloj. Esto permite funcionar a mayores frecuencias de reloj, pues puede reducirse la lógica que debe realizarse en cada ciclo, pero a cambio aumenta el tiempo total que tarda en realizarse una operación completa. En este caso además, por tratarse de un modelo en el que hay implementadas ecuaciones en diferencias existiendo por tanto una realimentación implícita, las operaciones se hacen con valores desactualizados de la salida anterior, lo que introduce un error adicional.

Se procede analizando el informe de tiempo que ofrece la herramienta ISE. Se introducen etapas de segmentación de forma que la división de la lógica sea tal que cada parte tarde aproximadamente lo mismo, de manera que no se tengan que utilizar más etapas de las estrictamente necesarias.

Con el procedimiento realizado, se llega a que se necesita un total de 5 etapas de registros para conseguir que se cumplan los requisitos de tiempos. Las señales PWM también deben ser registradas, pero esto no tiene ningún impacto en el error al estar funcionando en lazo abierto.

4.5.1.2 Modelo a 20 MHz

Como alternativa, se opta sencillamente por reducir la frecuencia de reloj a 20 MHz. Debido a que la frecuencia de conmutación del Buck sigue siendo de 200 kHz, habrá ahora únicamente 100 ciclos de cálculo por ciclo de conmutación, frente a los 250 que había al funcionar a 50 MHz. Hay que modificar por tanto el valor de cuenta que genera los PWMs desfasados. En lugar de contar hasta 249, y activar la señal de PWM en 5/12 del valor final (104 aproximadamente), se deberá contar hasta 99 activando la señal al llegar a 5/12 de ese valor: en el 42.

Los tiempos muertos introducen un problema adicional: antes duraban 2 ciclos de reloj, por lo que, para respetar la proporción, ahora deberían ser de $2/2,5 = 0.8$ ciclos. Como eso no es posible, se usan tiempos muertos de un único ciclo. Esos tiempos muertos no son proporcionales a los del modelo a 50 MHz, lo que introduce diferencias en el *duty cycle* del sistema cuando las corrientes de las bobinas son negativas, cosa que ocurre durante el régimen transitorio. Para tratar de analizar el rendimiento eliminando este problema, se realiza un nuevo modelo REAL funcionando a 20 MHz con los tiempos muertos de un ciclo, con el que poder comparar resultados.

En cuanto al modelo digitalizado del Buck, la única modificación que hay que llevar a cabo es cambiar el valor de las constantes `inct_L` e `inct_C`, pues aunque las constantes *L* y *C* no se han modificado, sí que se ha cambiado el paso de integración, que antes era de 20 ns y ahora de 50.

Para poder conseguir un reloj de ese período, debido a que el que existe en la placa Starter Kit Board es de 20 ns, se hace necesario un divisor de frecuencia. Para ello se utiliza un DCM que puede ser generado automáticamente por la herramienta de generación de IP Cores

del editor de Xilinx ISE. Se utiliza la salida CLKDV, indicando que el valor de división de frecuencia es de 5/2.

En el modelo REAL se utiliza un DCM de las mismas características. Esto se debe a que si se genera en el *testbench* directamente un reloj de 20 MHz, en lugar de generar uno de 50 y dividirlo, existen importantes diferencias en los resultados, debidas al tiempo que tarda el DCM en ofrecer una salida válida. Utilizando un divisor en ambos casos las diferencias desaparecen.

Comparando los resultados del modelo REAL funcionando a 20 MHz y del modelo en coma fija con la precisión optimizada, se observa que los resultados son bastante buenos, como se verá en el apartado 4.5.1.3.

4.5.1.3 Comparativa y selección de la solución

Ya se dispone de dos modelos que pueden ser sintetizados sobre la FPGA Spartan 3 y cumplen las restricciones de tiempo impuestas. Para evaluar cuál de las dos ofrece resultados más realistas, se comparan los valores que tienen sus tensiones de salidas y corrientes de cada fase en ciertos instantes de su régimen transitorio, con sus correspondientes modelos REAL, y los modelos de coma fija funcionando a 50 MHz sin modificaciones. Según puede verse en la Figura 37, los instantes elegidos coinciden con los picos de sobreoscilación de la tensión de los modelos. Aunque los valores de las señales son ligeramente diferentes en cada caso, los tiempos son exactamente iguales, de forma que en todos los casos el primer ciclo de sobreoscilación ocurre a los 110 μ s del inicio. Se elige además el instante 1800 μ s debido a que se desea conocer la diferencia que aparece en el valor final entre modelos, considerándose que en ese instante el régimen permanente ya ha sido alcanzado.

Las tablas Tabla 6, Tabla 7 y Tabla 8 ofrecen los datos de la comparativa realizada, en la que se han incluido, para más información, los modelos en coma fija a 50 MHz sin modificaciones tanto con precisión sin optimizar (*max precisión*) como optimizada (*opt precisión*).

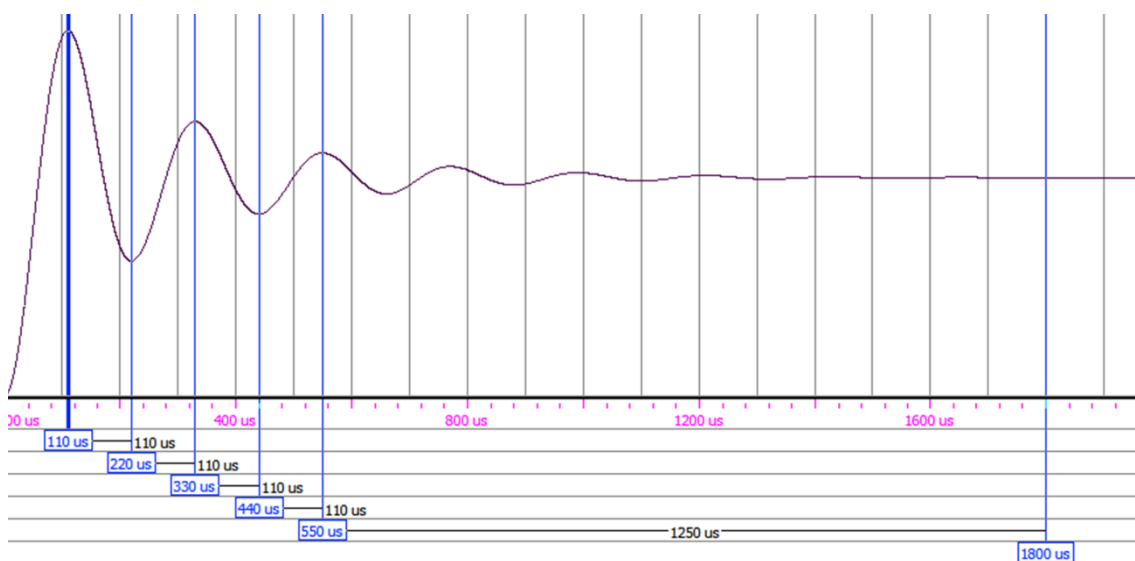


Figura 37: Régimen transitorio de la tensión de salida, con los instantes cuyos valores se comparan

Modelo	Sintentizable	110 us					220 us				
		Vout	IL1	IL2	IL3	IL4	Vout	IL1	IL2	IL3	IL4
Real 50 MHz	NO	8,41953	3,39162	3,33707	3,40253	3,40253	3,11672	1,21740	1,21740	1,30467	1,25012
Coma fija max precision 50 MHz	NO	8,41945	3,39182	3,33728	3,40273	3,40273	3,11684	1,21747	1,21747	1,30474	1,25020
Coma fija opt precision 50 MHz	NO	8,41945	3,39186	3,33723	3,40268	3,40271	3,11683	1,21745	1,21741	1,30476	1,25019
Coma fija opt precision 50 MHz pipeline	SI	8,43517	3,39737	3,31014	3,43004	3,43002	3,09817	1,21135	1,20034	1,32026	1,26574
Real 20 MHz	NO	8,42364	3,49188	3,30097	3,49188	3,49188	3,09374	1,18860	1,10678	1,29769	1,24315
Coma fija opt precision 20 MHz	SI	8,42356	3,49207	3,30116	3,49210	3,49208	3,09387	1,18865	1,10687	1,29779	1,24319

Tabla 6: Comparativa de modelos digitalizados (I)

Modelo	Sintentizable	330 us					440 us				
		Vout	IL1	IL2	IL3	IL4	Vout	IL1	IL2	IL3	IL4
Real 50 MHz	NO	6,32966	2,49736	2,49736	2,58463	2,53009	4,19653	1,58937	1,61119	1,81846	1,62210
Coma fija max precision 50 MHz	NO	6,32955	2,49510	2,49510	2,58479	2,53024	4,19662	1,58953	1,61115	1,81861	1,62234
Coma fija opt precision 50 MHz	NO	6,32956	2,49760	2,49746	2,58480	2,53034	4,19662	1,58947	1,61129	1,81856	1,62220
Coma fija opt precision 50 MHz pipeline	SI	6,34804	2,49777	2,48669	2,60667	2,55216	4,18589	1,57662	1,60930	1,84929	1,63106
Real 20 MHz	NO	6,34669	2,56804	2,48622	2,67713	2,62258	4,16282	1,62942	1,54760	1,73851	1,68396
Coma fija opt precision 20 MHz	SI	6,34657	2,56816	2,48640	2,67726	2,62268	4,16292	1,62947	1,54775	1,73866	1,68405

Tabla 7: Comparativa de modelos digitalizados (II)

Modelo	Sintentizable	550 us					1800 us				
		Vout	IL1	IL2	IL3	IL4	Vout	IL1	IL2	IL3	IL4
Real 50 MHz	NO	5,60562	2,17469	2,19651	2,40378	2,20742	5,03652	1,94458	1,96640	2,17367	1,97731
Coma fija max precision 50 MHz	NO	5,60553	2,17479	2,19641	2,40393	2,20740	5,03638	1,94474	1,96662	2,17418	1,97663
Coma fija opt precision 50 MHz	NO	5,60554	2,17482	2,19664	2,40392	2,20755	5,03652	1,94472	1,96653	2,17381	1,97744
Coma fija opt precision 50 MHz pipeline	SI	5,61537	2,16121	2,19402	2,43386	2,21560	5,03663	1,93286	1,96581	2,20429	1,98563
Real 20 MHz	NO	5,62894	2,26798	2,18616	2,37707	2,32252	5,03627	2,01696	1,93514	2,12605	2,07151
Coma fija opt precision 20 MHz	SI	5,62887	2,26801	2,18629	2,37724	2,32268	5,03638	2,01707	1,93576	2,12634	2,07148

Tabla 8: Comparativa de modelos digitalizados (III)

Los resultados dejan poco lugar a dudas: la solución consistente en bajar a 20 MHz la frecuencia de funcionamiento ofrece unos resultados mucho más cercanos a su modelo REAL a 20 MHz (y por tanto, previsiblemente, a la realidad) que los que ofrece el modelo a 50 MHz en el que se ha realizado un *pipeline* con respecto al modelo REAL a 50 MHz. En consecuencia, y dada la imposibilidad de sintetizar el modelo de 50 MHz sin modificaciones cumpliendo los requisitos de tiempo, se opta por el modelo de 20 MHz como solución definitiva.

CAPÍTULO 5: Integración final

Para poder cerrar el desarrollo del sistema, resta únicamente integrar los sistemas desarrollados, con ciertas pequeñas modificaciones para permitir la integración.

El modelo del convertidor Buck en coma fija es sintetizado sobre una FPGA Spartan 3, de forma que se mapean a los puertos de salida las señales que se necesitan para el control: tensión de salida y corrientes de cada fase. Las señales que se sacan al exterior son de 14 bits, debido a que el DAC de la placa de traducción tiene esa resolución. Como únicamente pueden representarse valores positivos, se saturan los valores negativos a 0 (únicamente las corrientes de cada fase, y durante el régimen transitorio, alcanzan valores negativos). Esto permite que el bit más significativo, que lleva la información de signo, sea descartado, y se añada sin embargo un nuevo bit por la parte menos significativa, lo que permite aumentar la precisión.

Como modificación adicional al sistema, se debe destacar la inclusión de un módulo anti-rebote que actúa sobre el botón de reseteo, de forma que se eviten problemas de oscilaciones al ser pulsado, problemas que causan perturbaciones en las señales y dificultan la observación de su régimen transitorio. El módulo anti-rebote se coloca directamente como un bloque de código en el *top level* del modelo digitalizado. Se puede ver su implementación en el Código 7. La señal de reseteo del sistema deja de ser asíncrona de forma que el reset se hace a través de la señal *resetX* de forma síncrona, evitándose posibles *glitches*.

```
--Proceso de debouncing de la señal de reset
counter_set <= flipflops(0) xor flipflops(1);

PROCESS(ClkDiv)
BEGIN
  IF(ClkDiv'EVENT and ClkDiv = '1') THEN
    flipflops(0) <= Reset;
    flipflops(1) <= flipflops(0);
    If(counter_set = '1') THEN
      counter_out <= (OTHERS => '0');
    ELSIF(counter_out(counter_size) = '0') THEN
      counter_out <= counter_out + 1;
    ELSE
      resetX <= flipflops(1);
    END IF;
  END IF;
END PROCESS;
```

Código 7: Implementación del módulo anti-rebote

Conectadas a las salidas del sistema se colocan las placas de conversión diseñadas, de forma que pueda obtenerse una señal analógica a partir de las señales digitales utilizadas. Conectando un osciloscopio a la salida del sistema, se observa el comportamiento real del modelo que se ha diseñado.

Ha de tenerse en cuenta la escala de las tensiones que se obtienen a la salida del sistema completo. Debido a que en todos los casos en la representación de salida se usan 4 bits para la parte entera y 10 para la parte decimal, ya que el MSB con la información de signo no se tiene en cuenta, una tensión de salida *del sistema completo* de 5 V (el máximo según el sistema desarrollado) se correspondería con un valor de la señal (tensión o corriente) de 1111,11111111 en coma fija sin signo, es decir, 15,9990234375 V. En consecuencia, un valor de aproximadamente 5 V en la tensión de salida *del modelo*, que es el esperado en régimen

permanente, es en coma fija 0101,0000000000, es decir $\frac{5}{15,999...} \cdot 5 \text{ V} = 1,5626 \text{ V}$. Se aprecia en la figura que ese es en efecto el valor de la tensión de salida del sistema en régimen permanente. De esta forma, cualquier valor a considerar como salida de un convertidor Buck real debe multiplicarse por $\frac{5}{15,999...} = 0,3125$ para saber el valor de tensión que saldrá de la placa de conversión.

Además, se observa que los picos y valles de sobreoscilación se producen en los instantes de tiempo que se esperaba según las simulaciones realizadas: el primer pico se produce aproximadamente 110 μs después del inicio (pulsado del botón de reseteado); el primer valle a 220 μs , el segundo pico a 330 μs ...

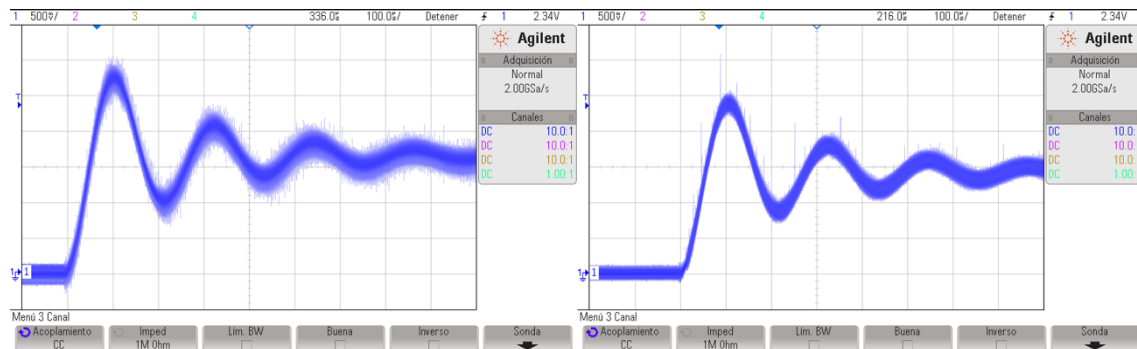


Figura 38: Transitorio de la tensión de salida con una corriente de salida de 8 A. Izquierda: placa conjunta. Derecha: placas separadas.

La Figura 38 muestra una comparativa en los transitorios de la tensión de salida con 8 A de corriente de salida, cuando se conecta a la salida la placa de conversión conjunta, o cuando por el contrario se utilizan placas separadas. La diferencia de nitidez se debe al efecto comentado anteriormente de no uniformidad en la longitud de las pistas. En el caso de las placas conjuntas, las diferencias de longitud de las pistas que debe recorrer cada señal es considerablemente mayor que en las placas separadas, entre otros factores debido a la mucho mayor longitud de las pistas y a que la longitud de los cables externos que unen FPGA y conector de la placa es menos uniforme en las conexiones placa conjunta-FPGA que en las conexiones placas separadas- FPGA (fundamentalmente, debido al conector empleado en cada caso). Es de esperar que este efecto se reduzca en gran medida cuando la placa conjunta sea conectada al conector de la SpCard para el que fue diseñada.

Una vez se ha comprobado el funcionamiento del sistema conjunto, se pasa a comparar con un convertidor Buck cuadrifase real, construido en [Usero]. La comparación requiere de varias modificaciones.

En primer lugar, se necesita sintetizar sobre una FPGA un generador de PWMs de ciclo de las mismas características que el que se conecta al modelo digitalizado del Buck, para poder conectarlo al Buck real de forma que funcione en lazo abierto.

Por otro lado, se hace necesario cambiar el condensador de salida que tenía el Buck real original, debido a que su elevada resistencia serie equivalente (ESR) introducía un rizado que modificaba considerablemente el régimen transitorio del sistema. Se pasa de un condensador de 220 μF a uno de 100 μF con mucha menor ESR. Esto supone que en el modelo digitalizado del convertidor también hay que modificar ese valor para poder comparar.

Finalmente, y dado que el Buck real se conecta a una carga cuyo valor puede cambiarse, se modifica el modelo de carga que se conecta al modelo digital del convertidor para que su valor pueda ser seleccionado utilizando dos *switches* de la FPGA. De esta forma, se permite seleccionar un valor de carga tal que la corriente de salida puede ser de 1 A ($5\ \Omega$), de 2,5 A ($2\ \Omega$), de 5 A ($1\ \Omega$) o de 8 A ($0,625\ \Omega$). La corriente de cada fase será aproximadamente una cuarta parte del valor de salida. Esto permite observar los efectos de la carga en el transitorio del sistema, y comparar el modelo digitalizado con el real en distintas situaciones.

Un ejemplo de la comparación se muestra a continuación para una carga de 0,625 ohmios. Si se desea, puede consultarse el ANEXO G para encontrar la comparación de régimen transitorio entre el modelo y la realidad con todos los posibles valores de carga considerados. En todos los casos, la comparación se realiza usando como módulo de conversión la placa conjunta, y debe tenerse en cuenta la transformación de escala mencionada anteriormente.

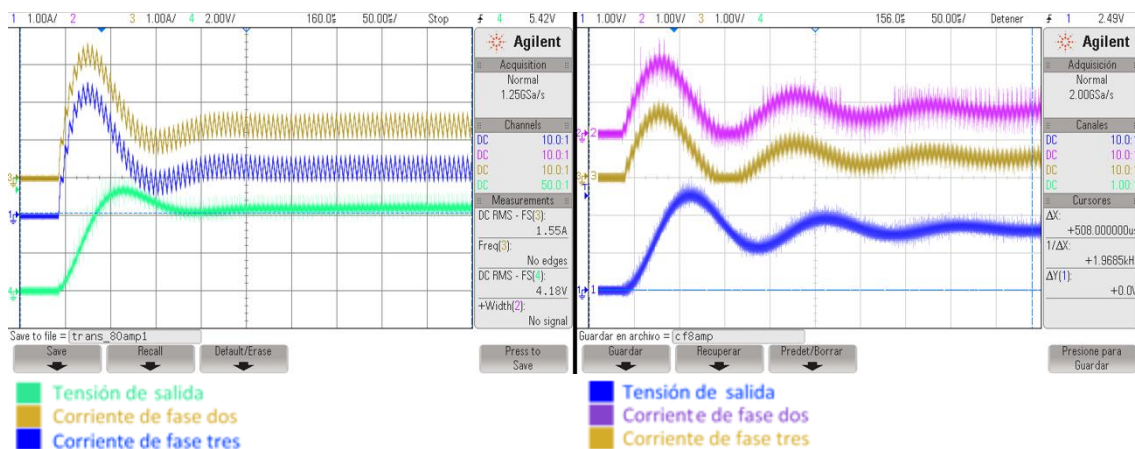


Figura 39: Comparativa de transitorios para corriente de salida de 8 A. Izquierda: Buck real. Derecha: Modelo HIL

Las aparentemente grandes diferencias entre el modelo y la realidad obedecen a una causa fundamental: las pérdidas. Pese a que los tiempos y valores en régimen permanente coinciden, no así los valores del transitorio: la influencia de las pérdidas que hay en el Buck real y no han sido introducidas en el modelo HIL hace que la sobre-oscilación sea en la realidad mucho menor de lo que se ha modelado.

En cualquier caso, se considera que la coincidencia de tiempos y del valor en régimen permanente es suficiente para avalar el buen funcionamiento del modelo diseñado. Si se consulta el ANEXO G, se observará que en ambos casos los cambios hacia cargas más bajas (mayores resistencias de salida) suponen sobreoscilaciones más pronunciadas, aunque este efecto es mucho más significativo en el modelo HIL, en el que las pérdidas no amortiguan las oscilaciones de la señal.

El motivo por el que las pruebas se realizan en lazo abierto es porque de esta forma se acentúan las diferencias entre modelos o entre modelo y Buck real. En lazo cerrado, el regulador se encargaría de llevar a cada señal a su valor de consigna aunque el modelo no fuera fidedigno, camuflando diferencias y fallos.

CAPÍTULO 6: Conclusiones y trabajo futuro

6.1. Conclusiones

Durante el presente trabajo se ha descrito la importancia que tiene en el proceso de prueba de un regulador el desarrollo de un modelo *Hardware in the Loop* (HIL) de la planta para la que se ha diseñado dicho regulador.

Una vez destacada la utilidad de este tipo de modelos, se ha desarrollado el proceso de diseño y construcción o síntesis de los dos módulos que integran la estructura habitual del modelo HIL de una planta analógica: modelo digitalizado de la planta y circuito de conversión digital a analógico. Dicho proceso de diseño constituye el cuerpo principal del trabajo que se presenta.

En el desarrollo del módulo de conversión se ha seguido un largo proceso de selección de componentes y configuraciones, en base a ciertos requisitos y pruebas, con el objetivo de optimizar su funcionamiento y permitir su uso con cualquier modelo digital con señales de salida de hasta 100 MHz. Se ha determinado que son necesarias dos etapas: una inicial de conversión digital a analógico, y una posterior de amplificación y acondicionamiento de la señal. Finalmente, como conversor digital a analógico de la primera etapa se ha elegido el modelo DAC904U, mientras que un amplificador operacional modelo CLC1005 funcionando en configuración no inversora ha ofrecido los mejores resultados para la etapa de amplificación. La unión de las dos etapas en una única placa ha permitido un diseño final mucho más compacto. Por otro lado, la inclusión de un conector tipo IDC de 24 pines y el rutado de las señales que llevan a él se ha realizado con el objetivo de permitir la compatibilidad de la placa de conversión con la de prototipado rápido de sistemas de control *SpCard*, en el marco de una colaboración del grupo de investigación *HCTLab* con la empresa *SpControl*.

Paralelamente, se ha desarrollado el modelo digital completo de un convertidor Buck de cuatro fases y frecuencia de conmutación de 200 kHz como planta analógica, resaltando la necesidad del uso de coma fija y síntesis sobre FPGA debida a la alta velocidad de las señales implicadas, de varios megahercios. El proceso ha comprendido varias fases, comenzando por el desarrollo de un diseño unifase utilizando señales de tipo REAL no sintetizables, para concluir con un diseño funcionando a 50 MHz en coma fija y resolución de señales optimizada sin pérdida de precisión significativa. Se ha explicado la imposibilidad de sintetizar el diseño a 50 MHz sobre una FPGA Spartan 3 sin realizar segmentación, y las posibles soluciones aplicables, concluyendo que la mejor posibilidad es la consistente en reducir la frecuencia de funcionamiento del sistema hasta 20 MHz.

Finalmente, se ha cerrado el trabajo con la integración de los dos módulos anteriores, constituyéndose un verdadero modelo HIL completo de un Buck cuadrifase con una frecuencia de conmutación de 200 kHz. Mediante el uso de un osciloscopio se han medido y comparado los transitorios del modelo HIL para diversas cargas con los transitorios de un convertidor Buck real con las mismas características. Pese a que las pérdidas provocan que la respuesta del Buck real sea, según se esperaba, mucho más amortiguada, la coincidencia de los tiempos en que se producen picos y valles de sobreoscilación, y del valor final permiten concluir que el modelo HIL desarrollado es perfectamente válido y funcional.

6.2. Trabajo futuro

Existen numerosas líneas de desarrollo futuro para el presente trabajo, fundamentalmente orientadas a la mejora de la integridad de señal de salida del módulo de conversión y al desarrollo de un modelo digital de la planta más realista y rápido.

Con respecto al primer objetivo, diseñar una placa de conversión conjunta en la que la tierra analógica y la digital estén separadas o incluso completamente desacopladas (por ejemplo, a través de opto-acopladores) es una buena forma de reducir el ruido. Utilizar las dos corrientes de salida del DAC elegido, usando una etapa de amplificación diferencial a continuación favorece asimismo una mejor integridad de señal, así como un mayor rango dinámico, a costa de la utilización de dos alimentaciones (positiva y negativa). Finalmente, se espera que la conexión de la placa a la SpCard para cuyo conector fue diseñada mejore significativamente la integridad de la señal resultante. En el marco de la colaboración entre el grupo HCTLab y SpControl se continuará el desarrollo y mejora de la placa, llevándose a cabo muchas de esas modificaciones, con el objetivo final de que la placa de traducción diseñada pueda convertirse en un producto comercial integrado en la SpCard.

Por otro lado, una de las posibles mejoras del modelo digital del Buck es incluir los efectos de las pérdidas, pues su impacto en el régimen transitorio del sistema es muy significativo. Finalmente, la implementación del formato de coma fija usando únicamente señales de tipo *std_logic_vector* en lugar de la biblioteca *sfixed* permitiría evitar ciertas operaciones de redondeo que requieren de un tiempo no despreciable sin un aumento destacable de precisión, constituyendo un importante progreso en materia de máxima frecuencia de funcionamiento.

CAPÍTULO 7: Bibliografía

- [AD9752ARZ] Analog Devices, *AD9752 (Rev A) Datasheet*
<http://www.analog.com/media/en/technical-documentation/data-sheets/AD9752.pdf>, 1999
- [AD9754ARZ] Analog Devices, *AD9754 (Rev A) Datasheet*
<http://www.analog.com/media/en/technical-documentation/data-sheets/AD9754.pdf>, 1999
- [ADA4857] Analog Devices, *ADA4857-1, ADA4857-2 (Rev. C) Datasheet*,
http://www.analog.com/media/en/technical-documentation/data-sheets/ADA4857-1_4857-2.pdf, 2008-2013
- [CLC1005] Exar Corporation, *CLC1005, CLC1015, CLC2005 Datasheet*,
<http://www.exar.com/common/content/document.ashx?id=21364>,
mar. 2015.
- [DAC904U] Texas Instruments, *DAC904 Datasheet*,
<http://www.ti.com.cn/cn/lit/ds/symlink/dac904.pdf>, 2002
- [dSPACE] dSPACE, <http://www.dspace.com/>, jun 2015
- [Farnell] Farnell, <http://es.farnell.com/>, jun 2015
- [HILSim] <http://www.hil-simulation.com/>, jun.2015
- [HCTLab] Human Computer Technology Laboratory, <http://www.hctlab.com/>,
jun. 2015.

[Jurado]	S. Jurado, “Emulación en FPGA del lazo cerrado de un controlador digital para convertidor conmutado”, Trabajo de Fin de Grado, sept. 2014
[Kester]	KESTER, Walt; BRYANT, James; BYRNE, Mike. “Grounding Data Converters and Solving the Mystery of AGND and DGND”. <i>Analog Devices Application Note</i> , 2006.
[Opal]	Opal-RT technologies, http://www.opal-rt.com/ , jun. 2015.
[RS]	RS Components, http://www.rs-components.com/ , jun 2015
[Sanchez]	A. Sánchez, “Aportaciones mediante implementación basada en sistemas embebidos al control digital de convertidores conmutados”, tesis doctoral, jun. 2013.
[SpControl]	Sp Control Technologies, http://www.spcontroltechnologies.com/ , jun. 2015.
[THS5651AIDW]	Texas Instruments, <i>THS5651AIDW Datasheet</i> http://www.ti.com/lit/ds/symlink/ths5651a.pdf , 2002
[Typhoon]	Typhoon HIL, https://www.typhoon-hil.com/ , jun 2015
[Usero]	L. Usero, “Diseño, implementación y control de reductor multifase”, Trabajo de Fin de Grado, sept. 2014
[Zumel]	P. Zumel, M. García-Valderas, A. Lázaro, C. Loópez-Ongil and A. Barrado, “Co-simulation psim-modelsim oriented to digitally controlled switching power converters” in <i>Control and Modeling for Power Electronics (COMPEL)</i> , 2010 IEEE 12 th Workshop on, pp. 1-7, jun. 2010.

ANEXO A Pinout y huella de los componentes

Conversor digital-analógico AD9752ARZ

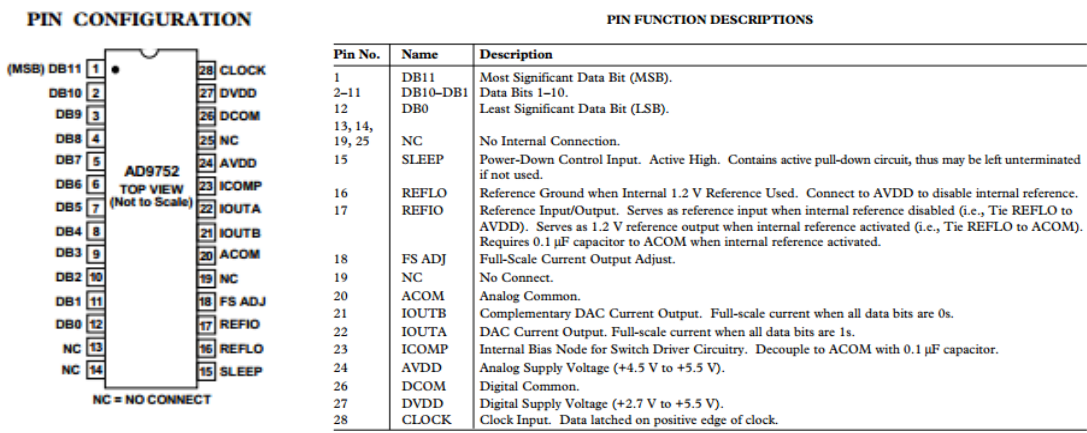


Figura 40: Pinout y huella del AD9752ARZ

Conversor digital-analógico AD9754ARZ

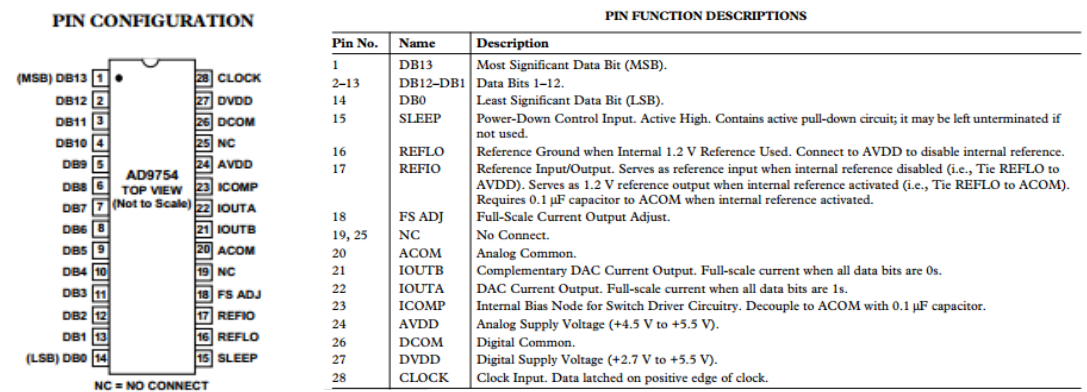


Figura 41: Pinout y huella del AD9754ARZ

Conversor digital-analógico DAC904U

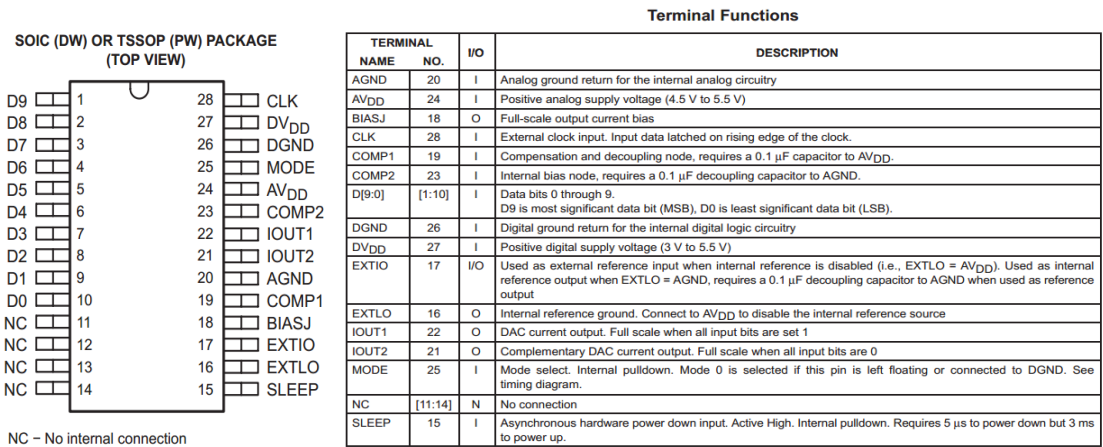
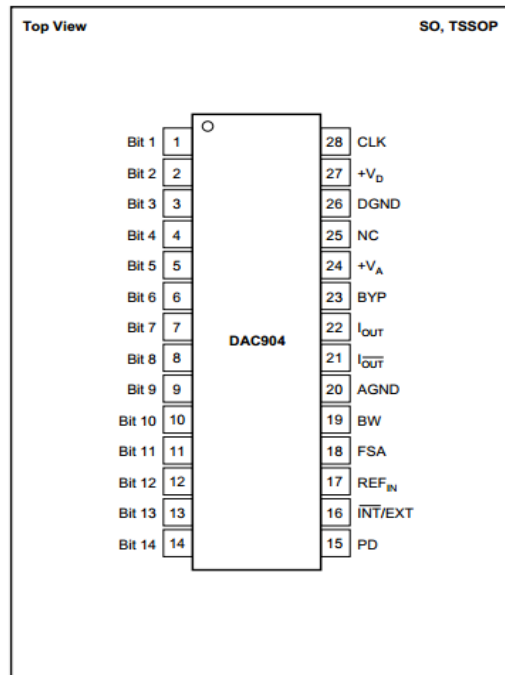


Figura 42: Pinout y huella del AD9752ARZ

Conversor digital-analógico THS561AIDW

PIN CONFIGURATION



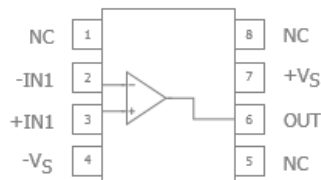
PIN DESCRIPTIONS

PIN	DESIGNATOR	DESCRIPTION
1	Bit 1	Data Bit 1 (D13), MSB
2	Bit 2	Data Bit 2 (D12)
3	Bit 3	Data Bit 3 (D11)
4	Bit 4	Data Bit 4 (D10)
5	Bit 5	Data Bit 5 (D9)
6	Bit 6	Data Bit 6 (D8)
7	Bit 7	Data Bit 7 (D7)
8	Bit 8	Data Bit 8 (D6)
9	Bit 9	Data Bit 9 (D5)
10	Bit 10	Data Bit 10 (D4)
11	Bit 11	Data Bit 11 (D3)
12	Bit 12	Data Bit 12 (D2)
13	Bit 13	Data Bit 13 (D1)
14	Bit 14	Data Bit 14 (D0), LSB
15	PD	Power Down, Control Input: Active HIGH. Contains internal pull-down circuit; may be left unconnected if not used.
16	INT/EXT	Reference Select Pin; Internal (= 0) or External (= 1) Reference Operation
17	REF _{IN}	Reference Input/Output. See Applications section for further details.
18	FSA	Full-Scale Output Adjust
19	BW	Bandwidth/Noise Reduction Pin: Bypass with 0.1μF to +V _A for Optimum Performance. (Optional)
20	AGND	Analog Ground
21	I _{OUT}	Complementary DAC Current Output
22	I _{OUT}	DAC Current Output
23	BYP	Bypass Node: Use 0.1μF to AGND
24	+V _A	Analog Supply Voltage, 2.7V to 5.5V
25	NC	No Internal Connection
26	DGND	Digital Ground
27	+V _D	Digital Supply Voltage, 2.7V to 5.5V
28	CLK	Clock Input

Figura 43: Pinout y huella del THS561AIDW

Amplificador operacional CLC1005

SOIC-8



SOIC-8

Pin No.	Pin Name	Description
1	NC	No Connect
2	-IN	Negative input
3	+IN	Positive input
4	-V ₅	Negative supply
5	NC	No Connect
6	OUT	Negative input
7	+V ₅	Positive supply
8	NC	No Connect

Figura 44: Pinout y huella del CLC1005

Amplificador operacional ADA4857



Pin No.	Mnemonic	Description
1	FB	Feedback.
2	-IN	Inverting Input.
3	+IN	Noninverting Input.
4	-V ₅	Negative Supply.
5	NC	No Connect.
6	OUT	Output.
7	+V ₅	Positive Supply.
8	PD	Power Down.

Figura 45: Pinout y huella del ADA4857

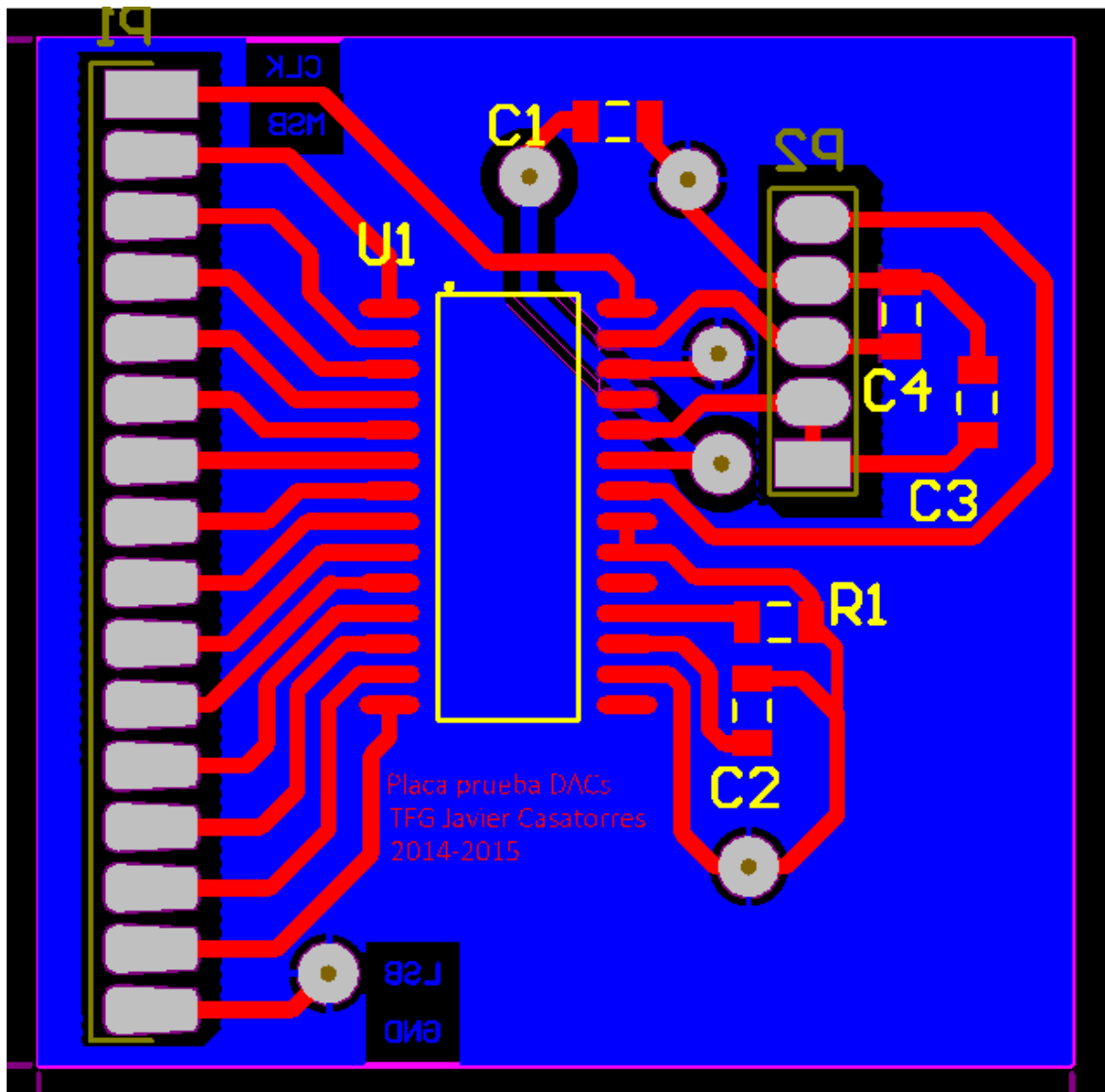


Figura 47: Layout de la placa de conversión común

Placa de conversión THS5651AIDW

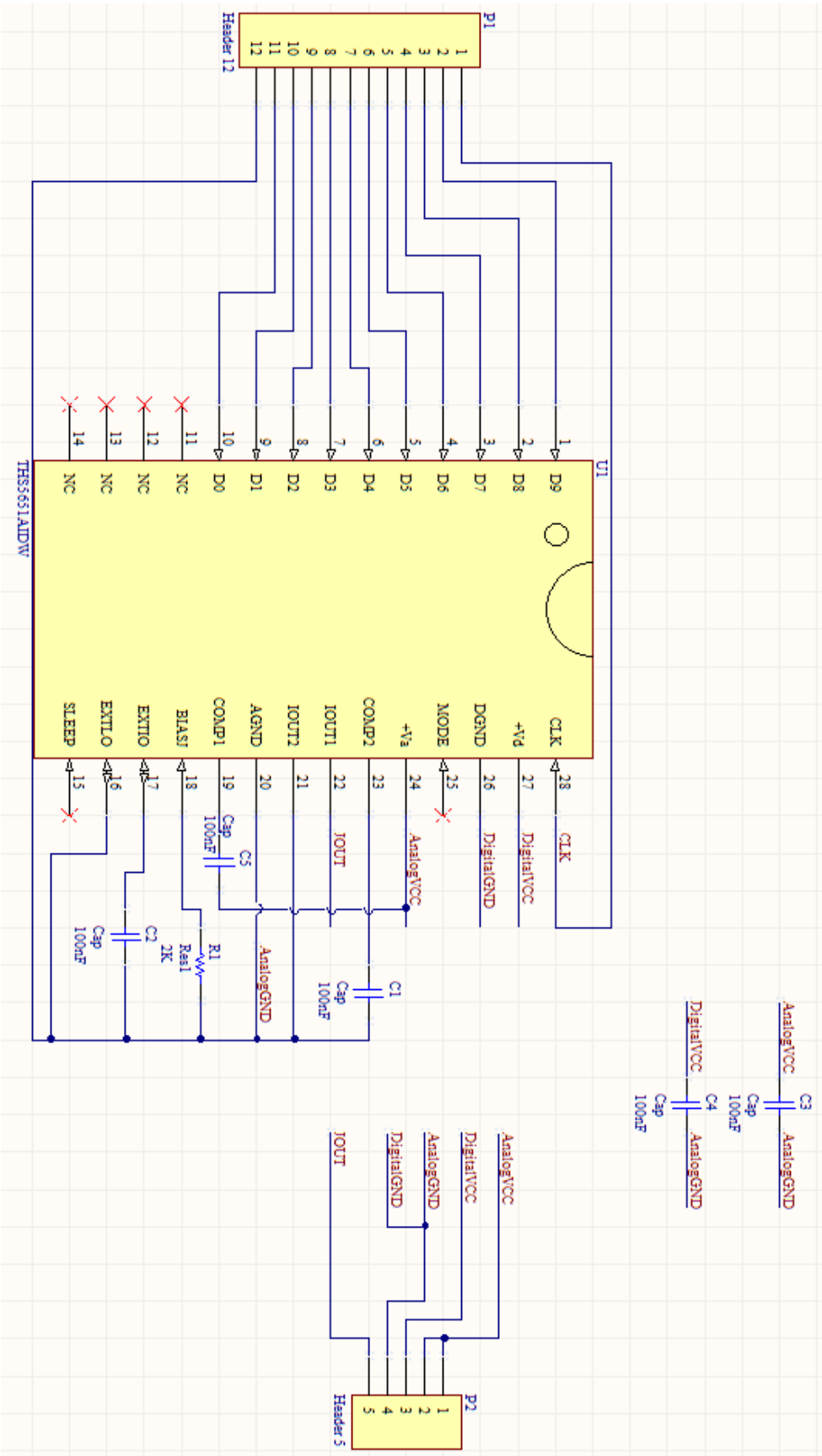


Figura 48: Esquemático de la placa de conversión del DAC THS5651AIDW

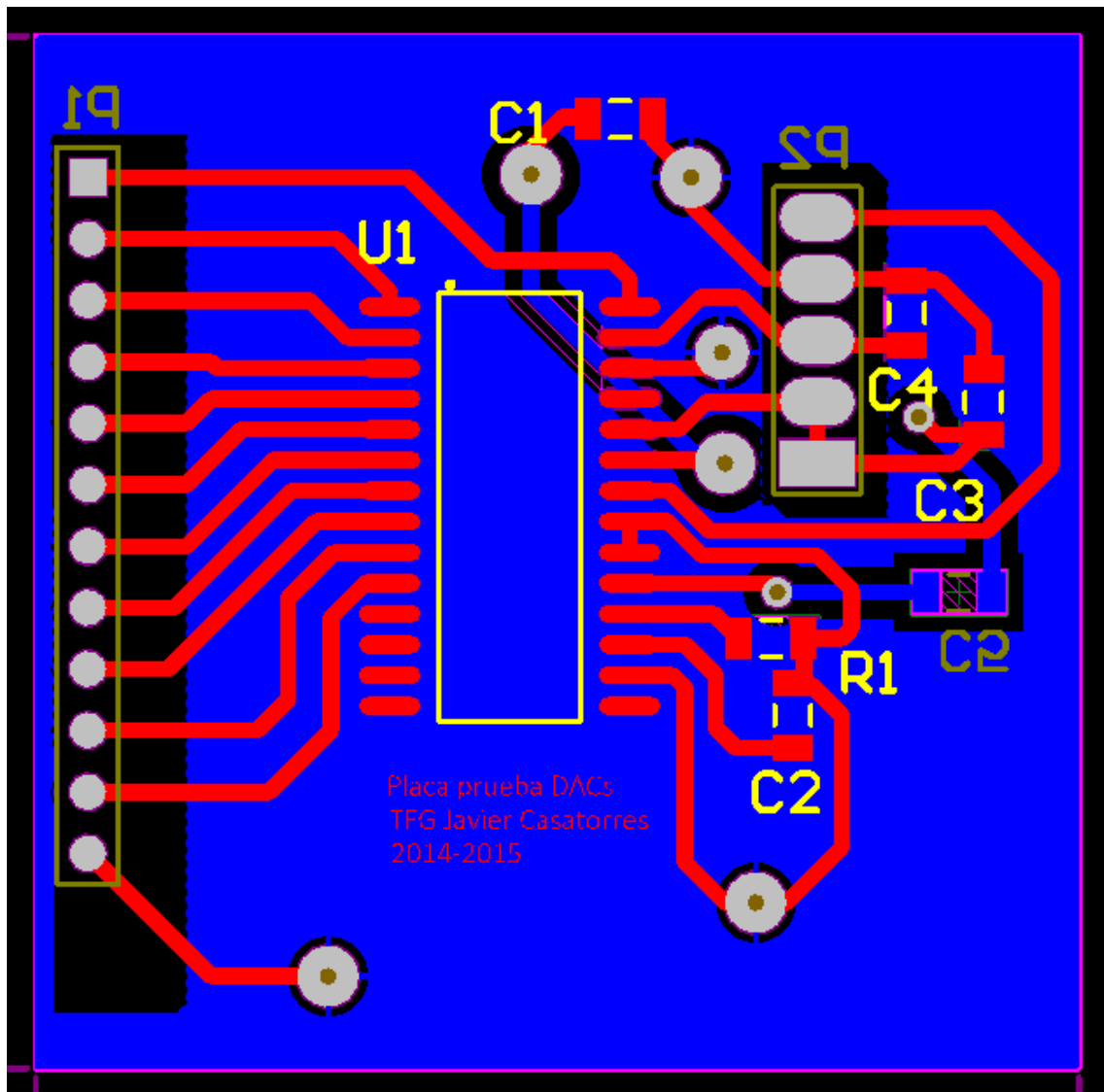


Figura 49: Layout de la placa de conversión del DAC THS5651AIDW

Placa de amplificación inversora

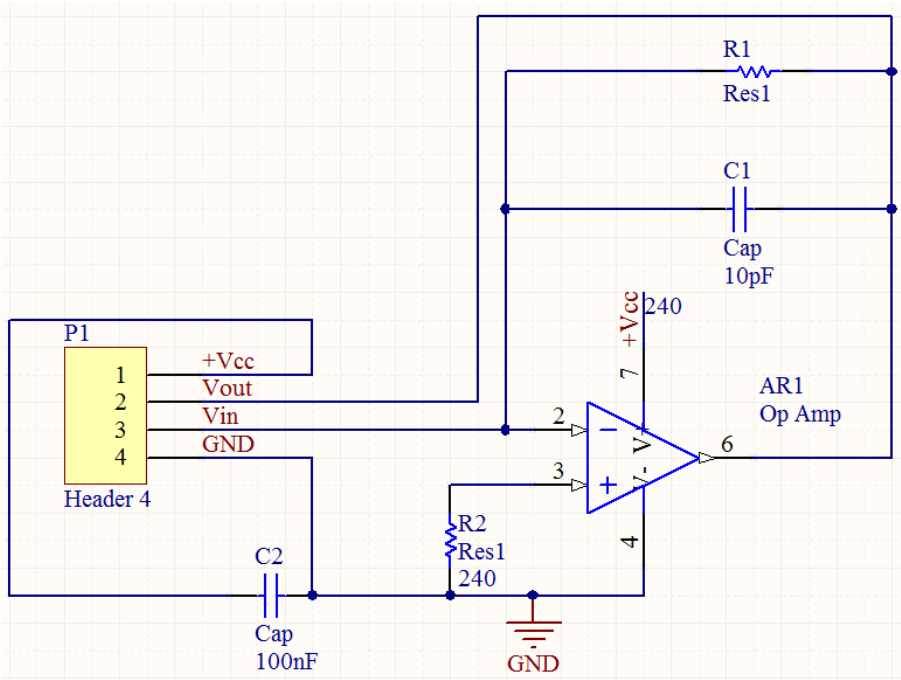


Figura 50: Esquemático de la etapa de amplificación en configuración no inversora

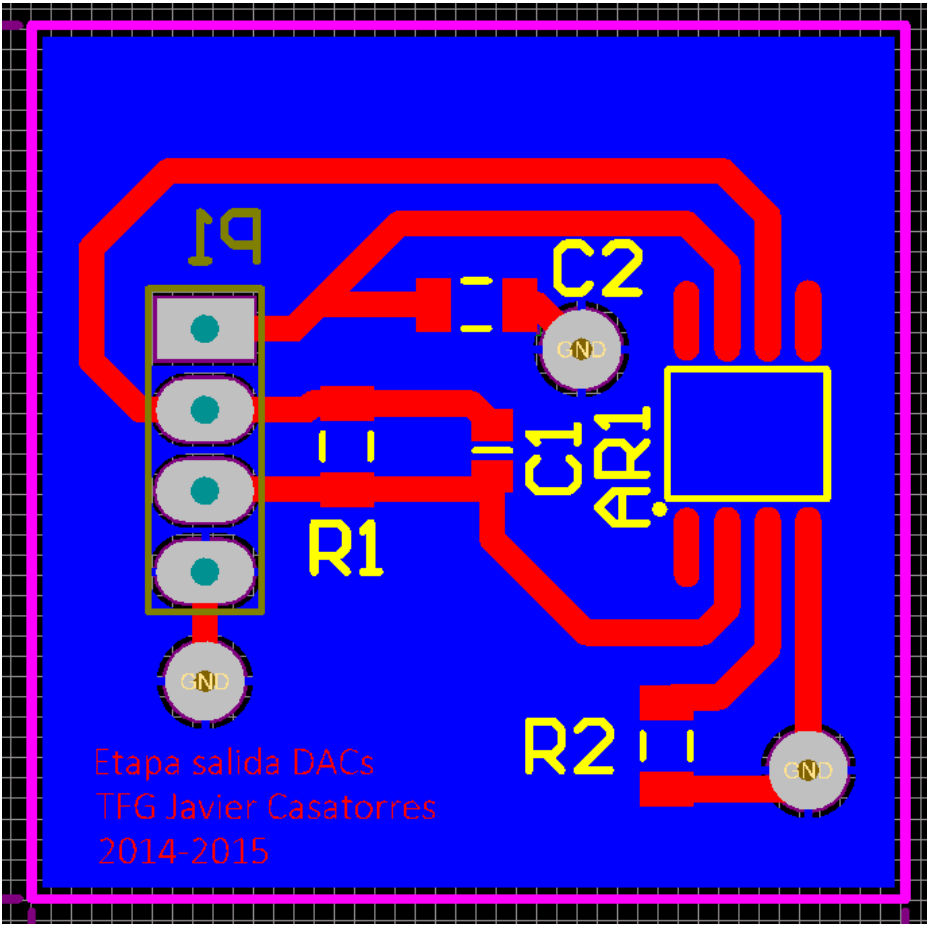


Figura 51: Layout de la etapa de amplificación en configuración no inversora

Placa de amplificación no inversora

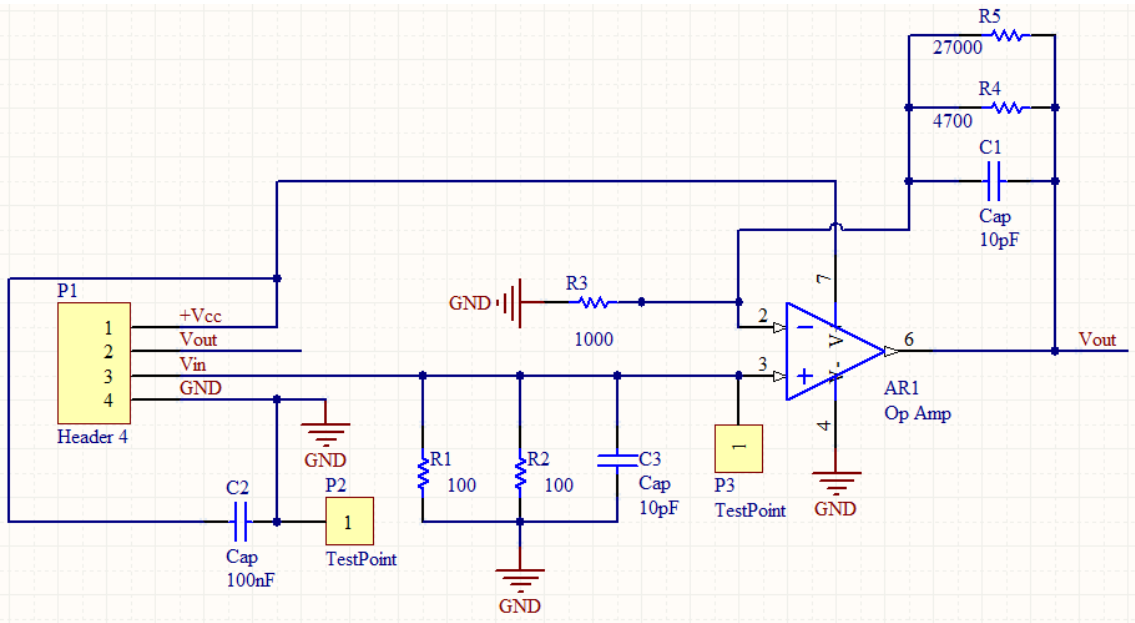


Figura 52: Esquemático de la placa de amplificación en configuración no inversora

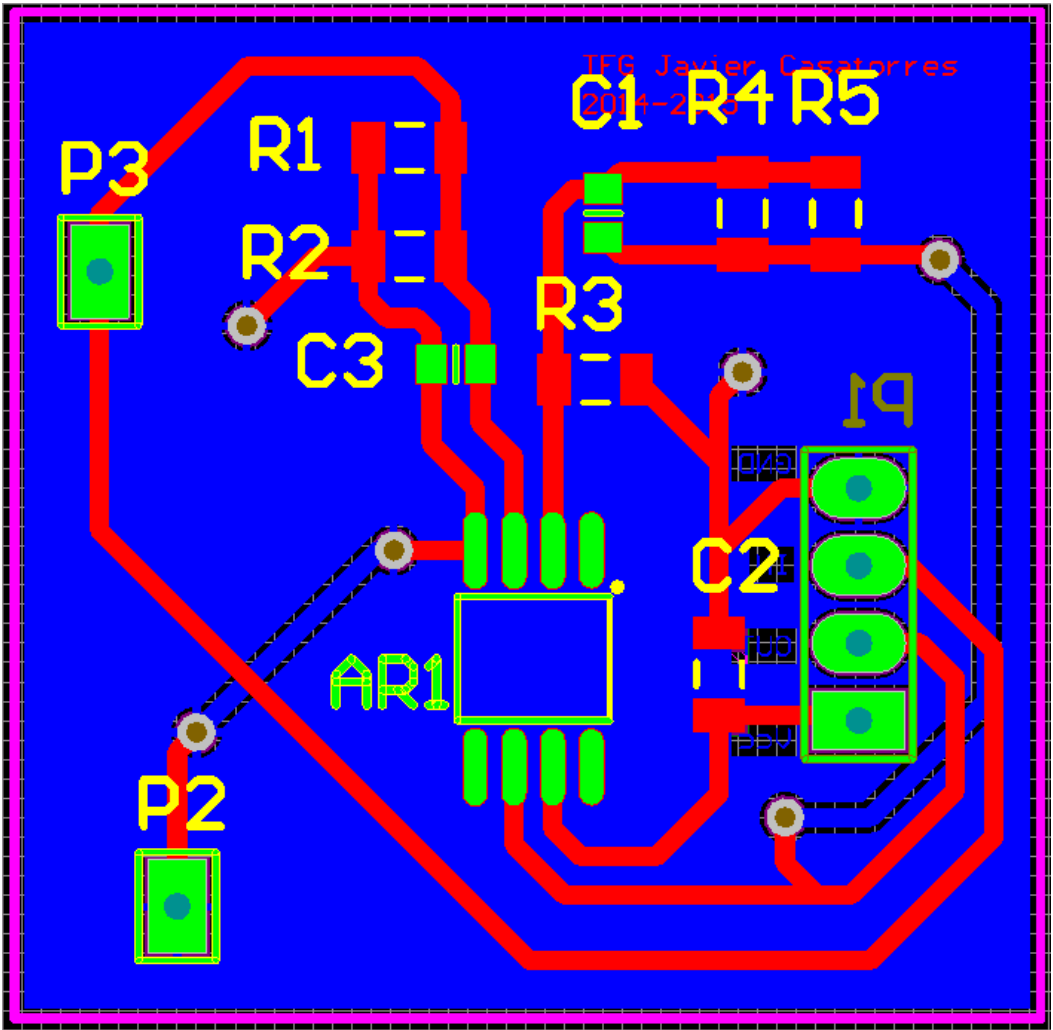


Figura 53: Layout de la placa de amplificación en configuración no inversora

Placas conjuntas

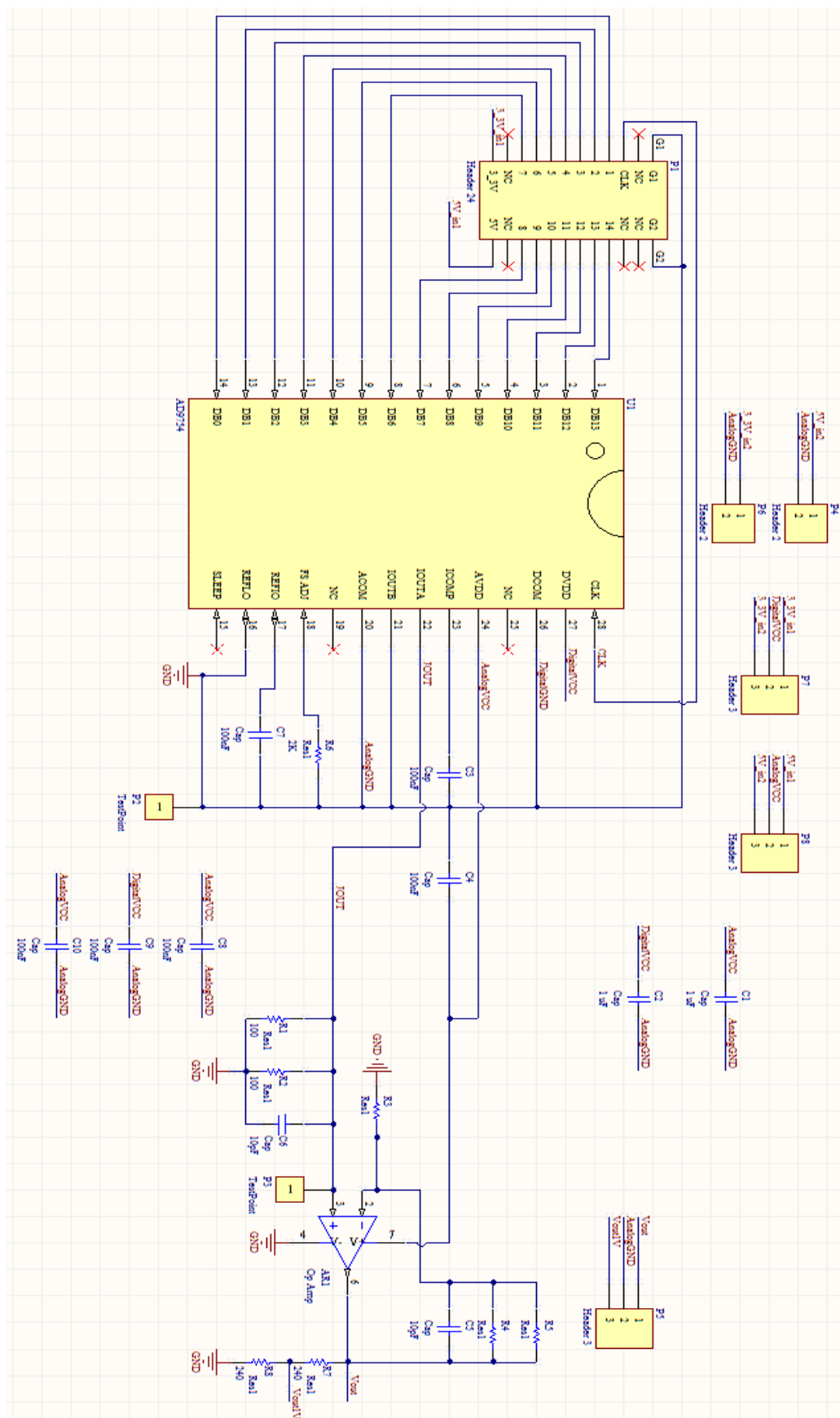


Figura 54: Esquemático de las placas conjuntas

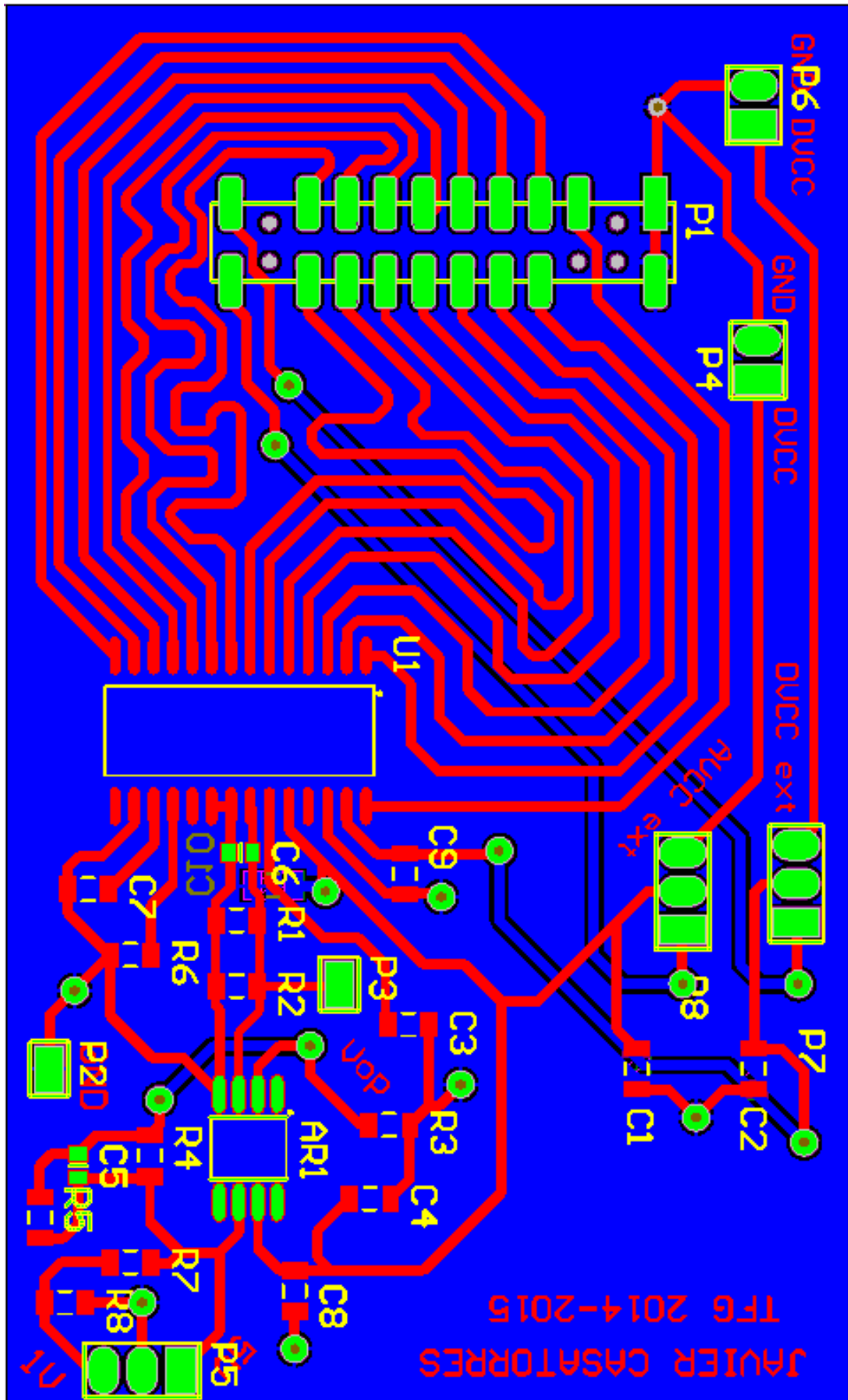


Figura 55: Layout de la placa con junta con uniformización de longitud de pistas

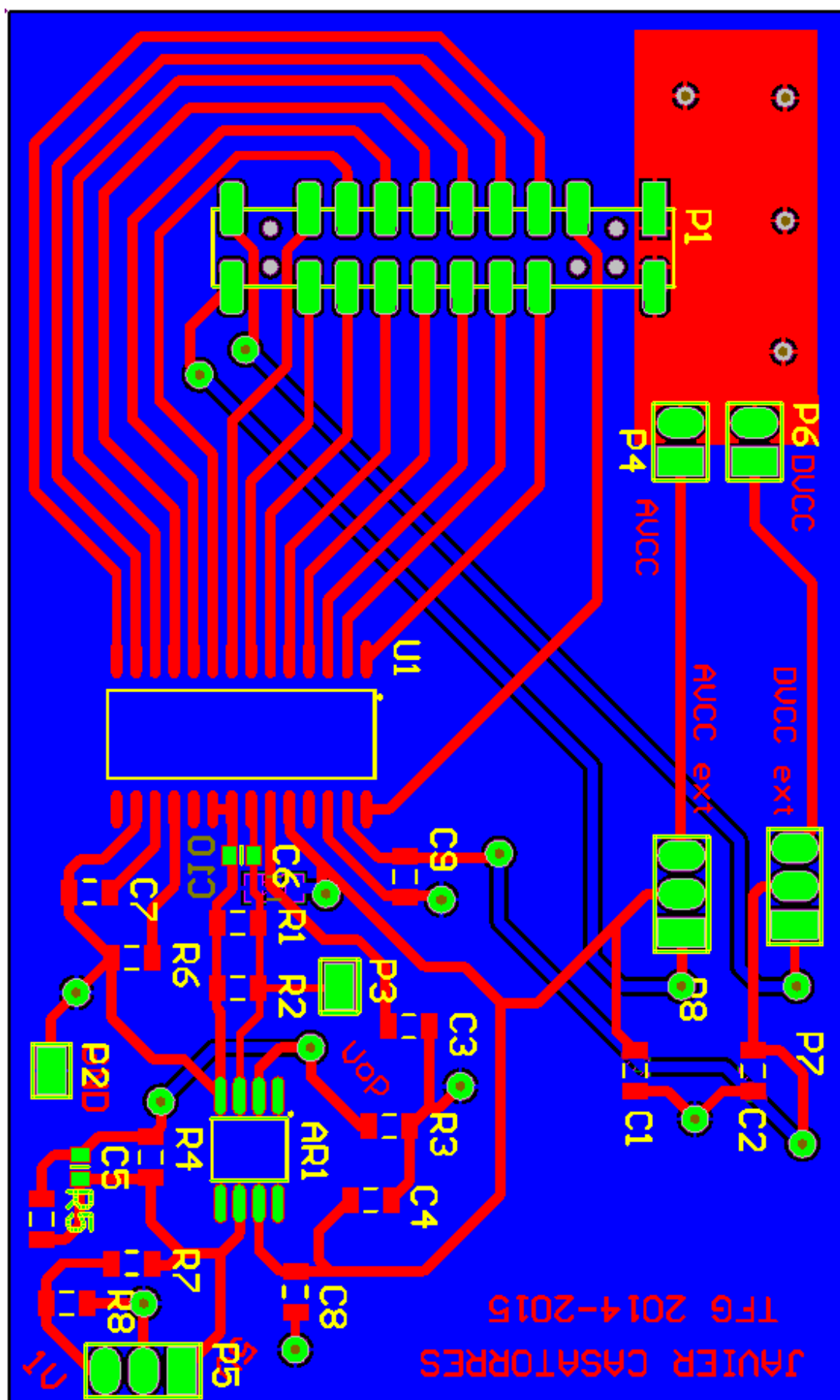


Figura 56: Layout de la placa conjunta sin uniformización de pistas

ANEXO C Generadores de señales de prueba

Generador de escalones

```
-----
--      Generador de escalones
--      TFG Javier Casatorres Agüero
--      2014-2015
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Botones is
  Port (
    DataIN : in STD_LOGIC_VECTOR (7 downto 0);
    DataOUT : out STD_LOGIC_VECTOR (15 downto 0);
    LEDsOUT : out STD_LOGIC_VECTOR (7 downto 0);
    CLKPortOut : out STD_LOGIC;
    CLKPort : in STD_LOGIC);
end Botones;

architecture Behavioral of Botones is
begin
  --Los MSB de la salida vienen de los switches
  --Los LSB de la salida se ponen a 1
  DataOUT(15 downto 8) <= DataIN;
  DataOUT(7 downto 0) <= "00000000";
  --Asignamos los LEDs a a los pulsadores para depurar
  LEDsOUT <= DataIN;
  --Asignamos el reloj
  CLKPortOut <= CLKPort;
end Behavioral;
```

Código 8: Generador de señales escalón

Generador de señales

```
-----
--      Generador de señales
--      TFG Javier Casatorres Agüero
--      2014-2015
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.ALL;

entity TopLevelControlMemoria is
  Port ( Reset :in STD_LOGIC;
    OutDAC : out STD_LOGIC_VECTOR (15 downto 0);
    CLKport : in STD_LOGIC;
    CLKportOut : out STD_LOGIC);
end TopLevelControlMemoria;

architecture Behavioral of TopLevelControlMemoria is

  -- DEFININION DE SEÑALES --
  signal datosOut: std_logic_vector(15 downto 0) := X"0000";
  signal direccion: std_logic_vector(9 downto 0) := "0000000000";
  signal clk: std_logic;
```

```

-- DECLARACION DE MEMORIA--
component dc_mem
  port (
    signal DO : out std_logic_vector(15 downto 0);      -- 16-bit Data Output
    signal ADDR :in std_logic_vector(9 downto 0);       -- 10-bit Address Input
    signal CLK :in std_logic;                           -- Clock
    signal EN :in std_logic;                            -- RAM Enable Input
    signal SSR :in std_logic;                          -- Synchronous Set/Reset Input
    signal WE :in std_logic                            -- Write Enable Input
  );
end component;

-----

begin

-- INSTANCIACION DE MEMORIA--
memoria : dc_mem
  port map (
    DO => datosOut,
    ADDR => direccion,
    CLK => clk,
    EN => '1',
    SSR => reset,
    WE => '0'
  );

  clk<=CLKport;
  CLKportOut<=not clk;
  OutDAC<=datosOut;

process (CLK,reset)
begin
  if(reset='1') then
    direccion<=(others=>'0');
  elsif CLK'event and CLK='1' then
    if(direccion = "1111100111") then --Cuando llega a 1000 se reinicia: 1000 direcciones
      direccion <= (others=>'0');
    else
      direccion <= std_logic_vector(unsigned(direccion)+1);
    end if;
  end if;
end process;
end Behavioral;

```

Código 9: Generador de señales

Asignación de pines

```
NET "CLKport" TNM_NET = CLKport;
TIMESPEC TS_CLKport = PERIOD "CLKport" 20 ns HIGH 50%;

NET "OutDAC<15>" LOC = B14;
NET "OutDAC<14>" LOC = B13;
NET "OutDAC<13>" LOC = B12;
NET "OutDAC<12>" LOC = A10;
NET "OutDAC<11>" LOC = A9;
NET "OutDAC<10>" LOC = B8;
NET "OutDAC<9>" LOC = B7;
NET "OutDAC<8>" LOC = A5;
NET "OutDAC<7>" LOC = A4;
NET "OutDAC<6>" LOC = A3;
NET "OutDAC<5>" LOC = C9;
NET "OutDAC<4>" LOC = C8;
NET "OutDAC<3>" LOC = C7;
NET "OutDAC<2>" LOC = C6;
NET "OutDAC<1>" LOC = C5;
NET "OutDAC<0>" LOC = E6;

NET "Reset" LOC = L14;
NET "CLKport" LOC = T9;
NET "CLKportOut" LOC = D5;
```

Código 10: Asignación de pines

ANEXO D Pruebas con señales

Seno a 50 kHz

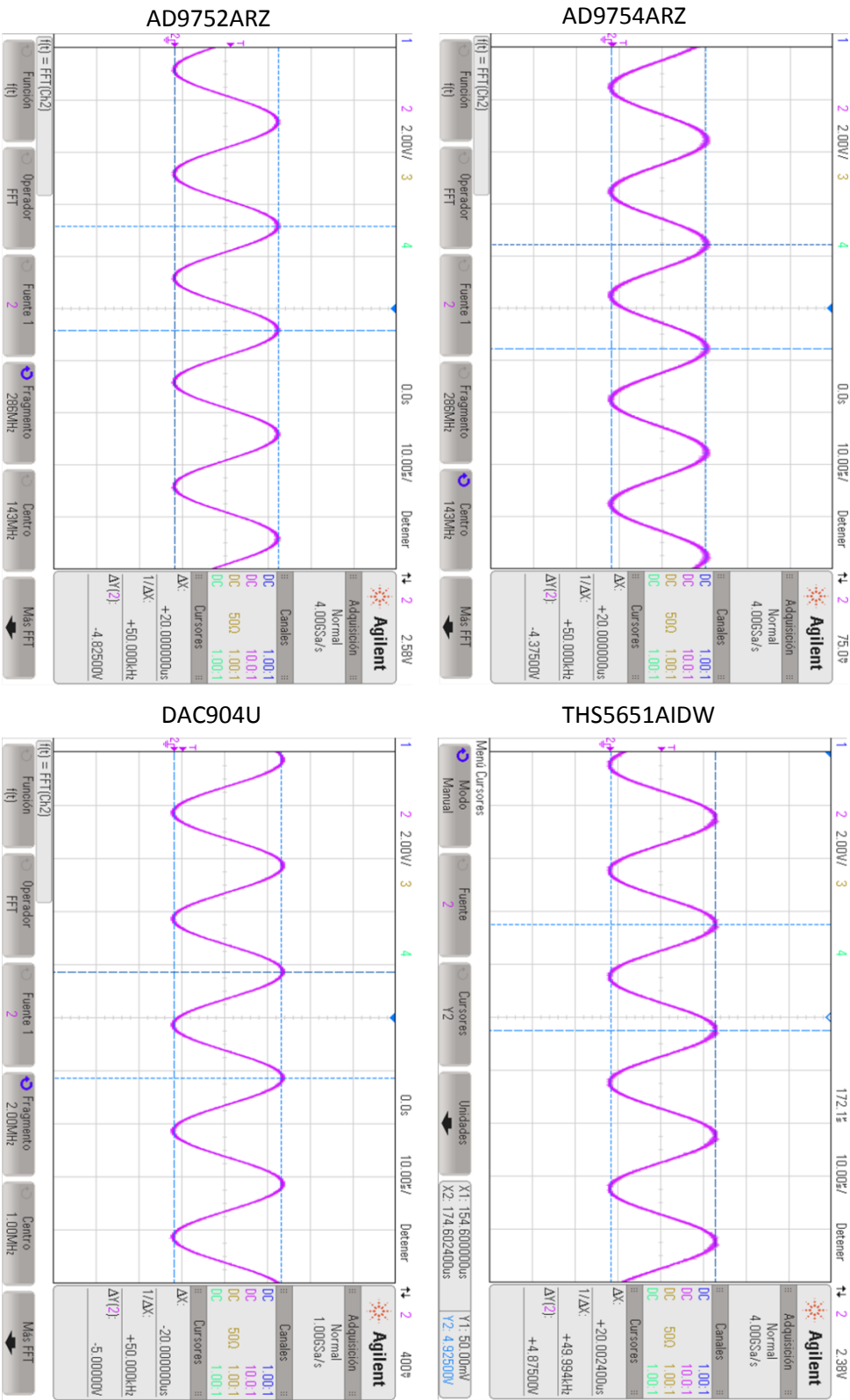


Figura 57: Seno en el tiempo a 50 kHz

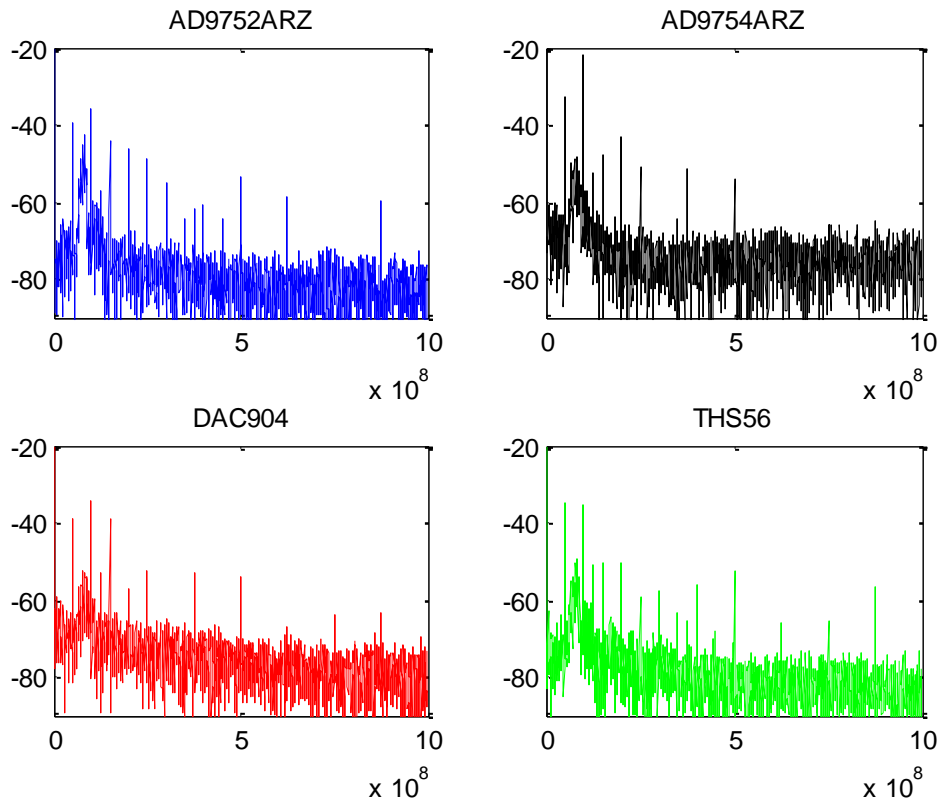


Figura 58: Espectro del seno a 50 kHz en banda ancha

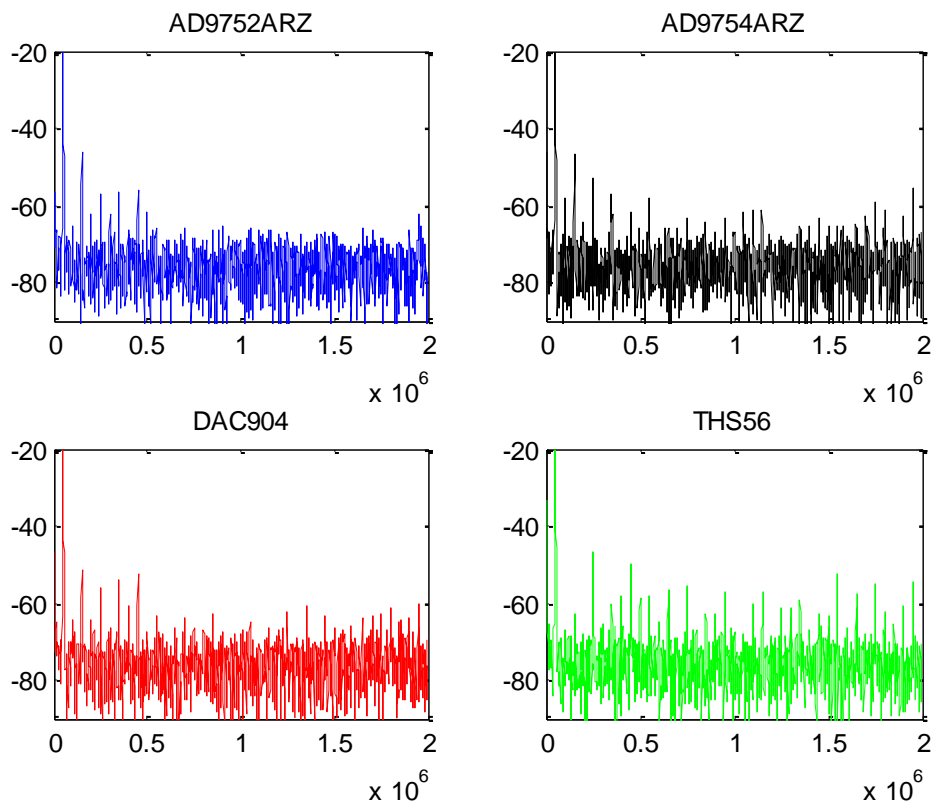


Figura 59: Espectro del seno a 50 kHz en banda estrecha

Seno a 1 MHz

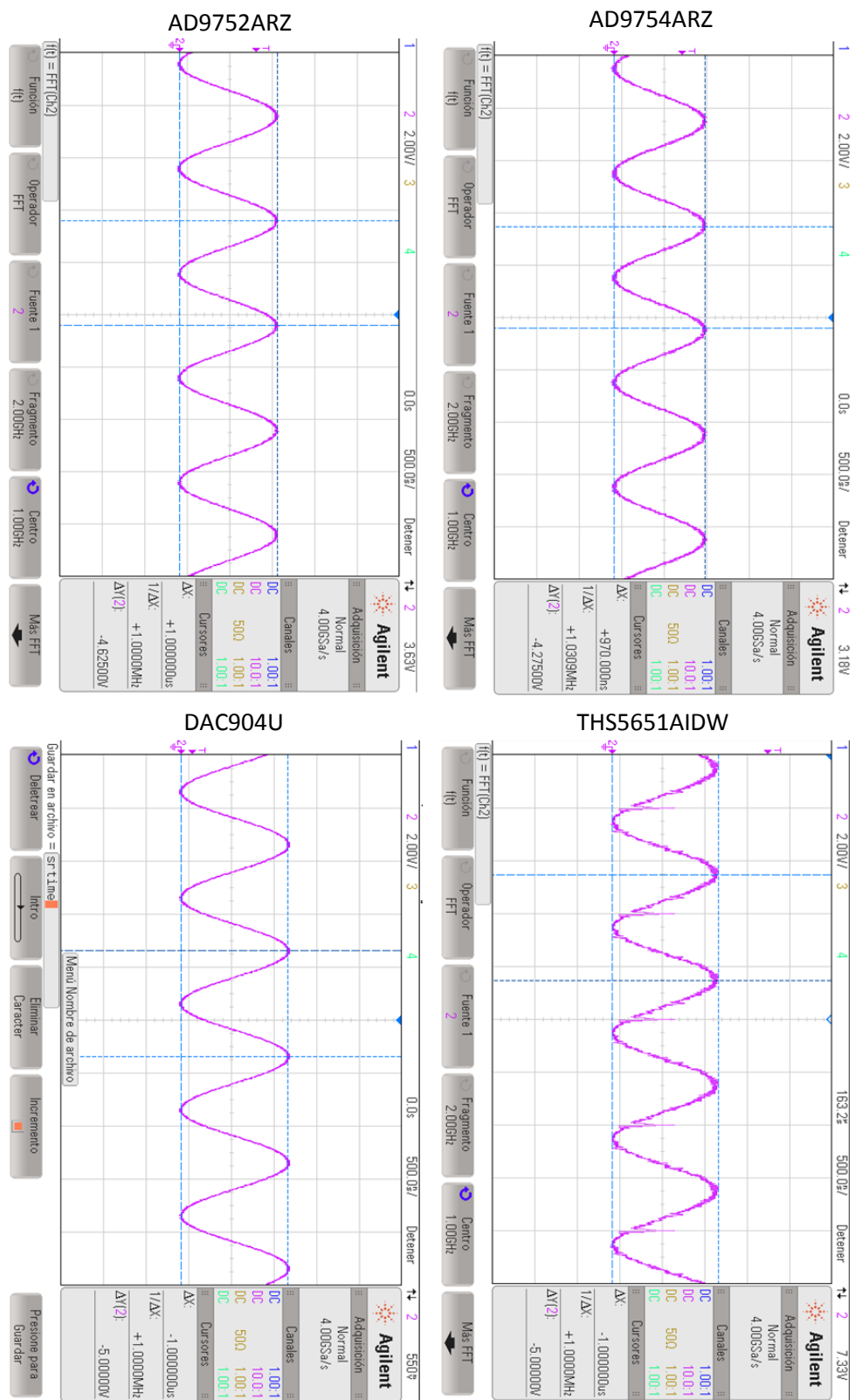


Figura 60: Seno en el tiempo a 1 MHz

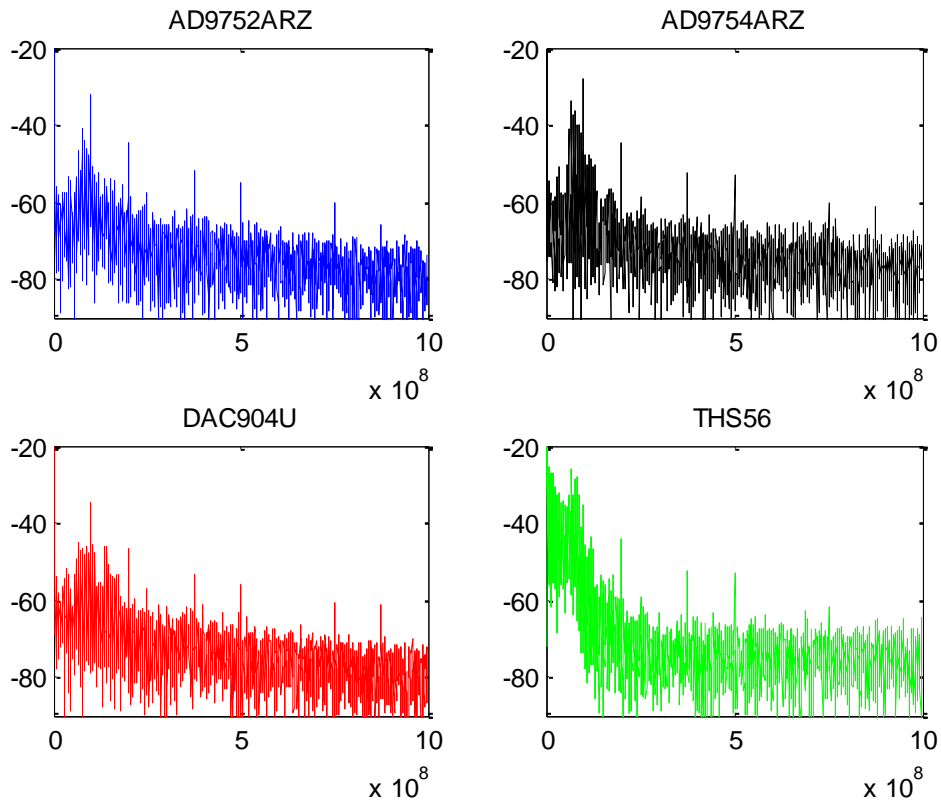


Figura 61: Espectro del seno a 1 MHz en banda ancha

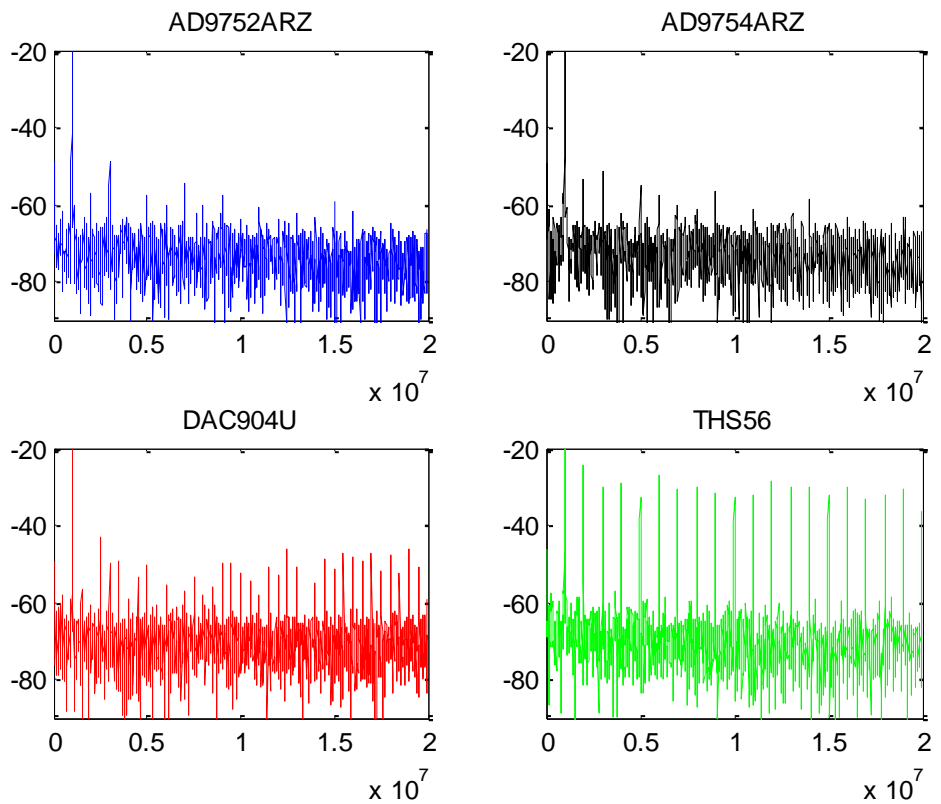


Figura 62: Espectro del seno a 1 MHz en banda estrecha

Triangular a 50 kHz

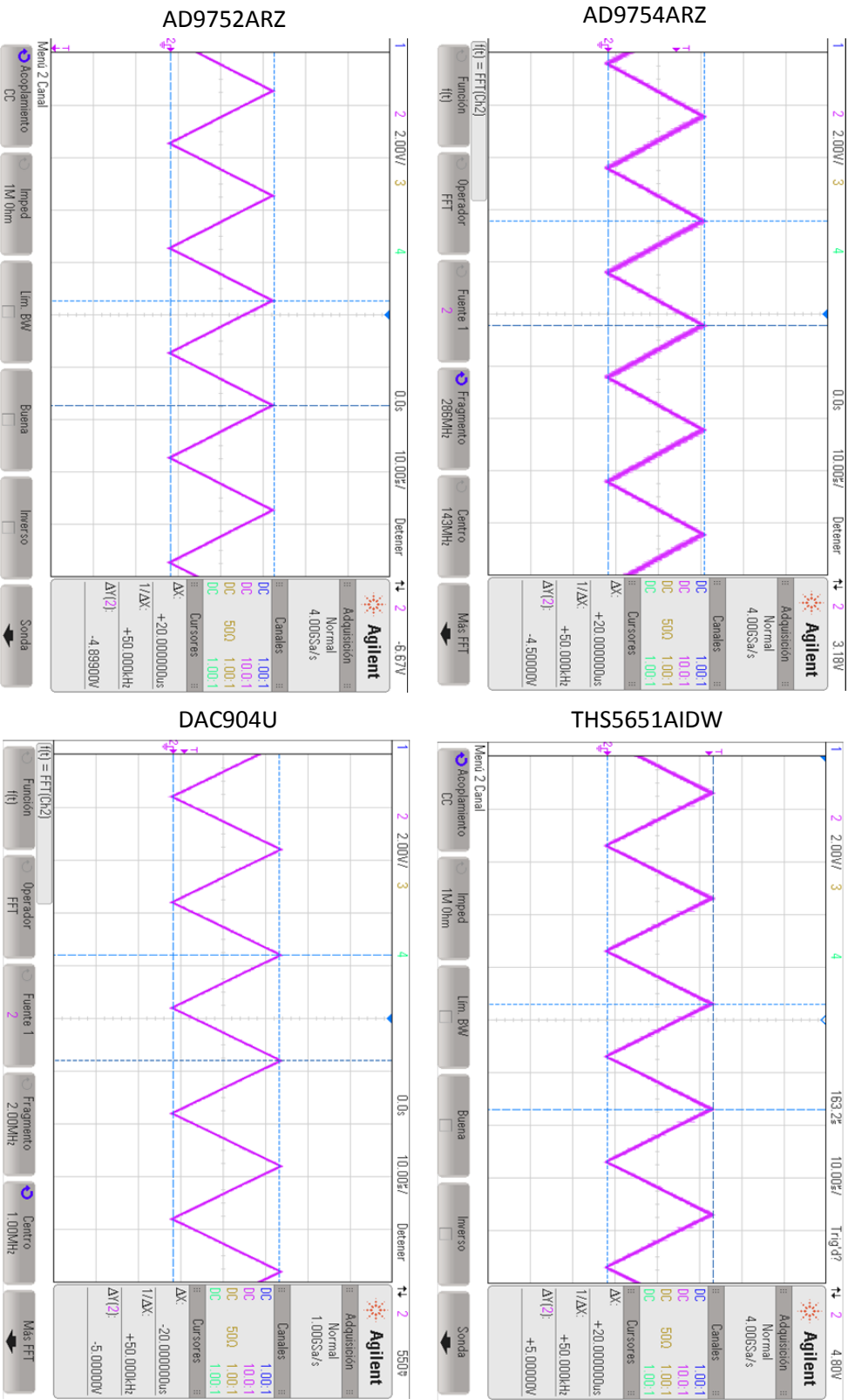


Figura 63: Triangular en el tiempo a 50 kHz

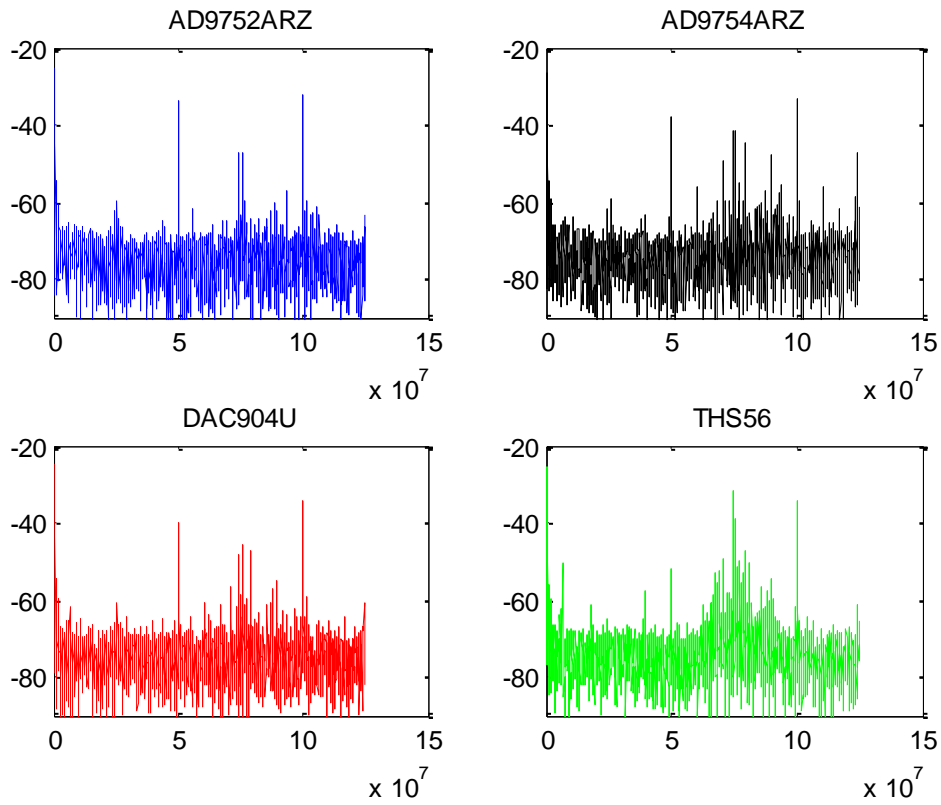


Figura 64: Espectro de señal triangular de 50 kHz en banda ancha

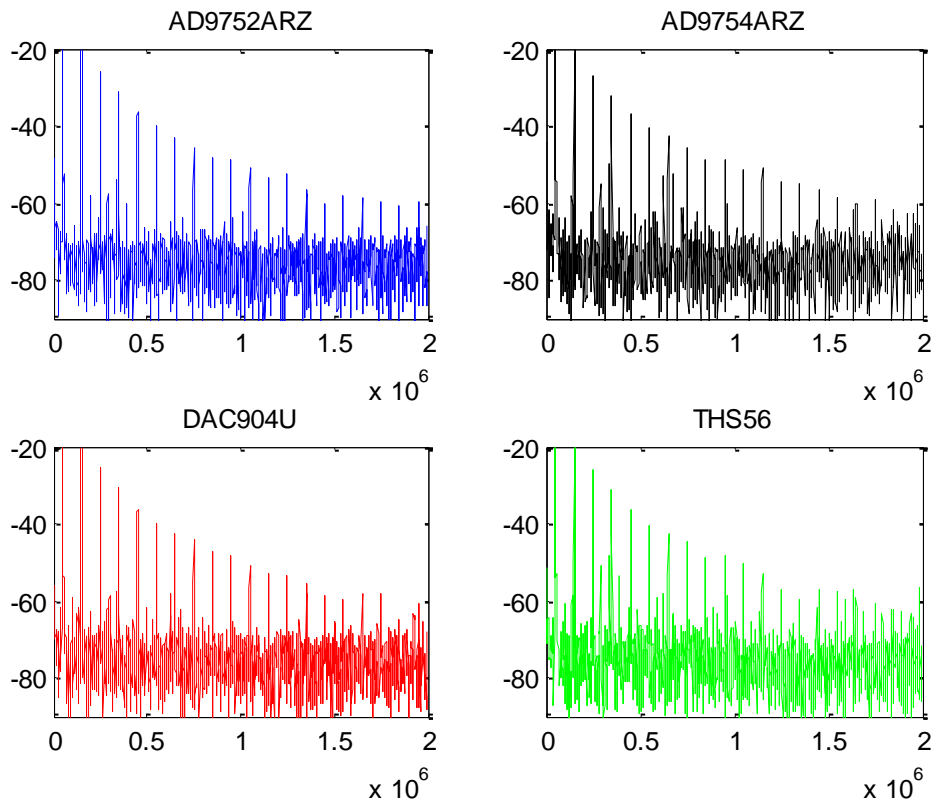


Figura 65: Espectro de señal triangular de 50 kHz en banda estrecha

Triangular a 1 MHz

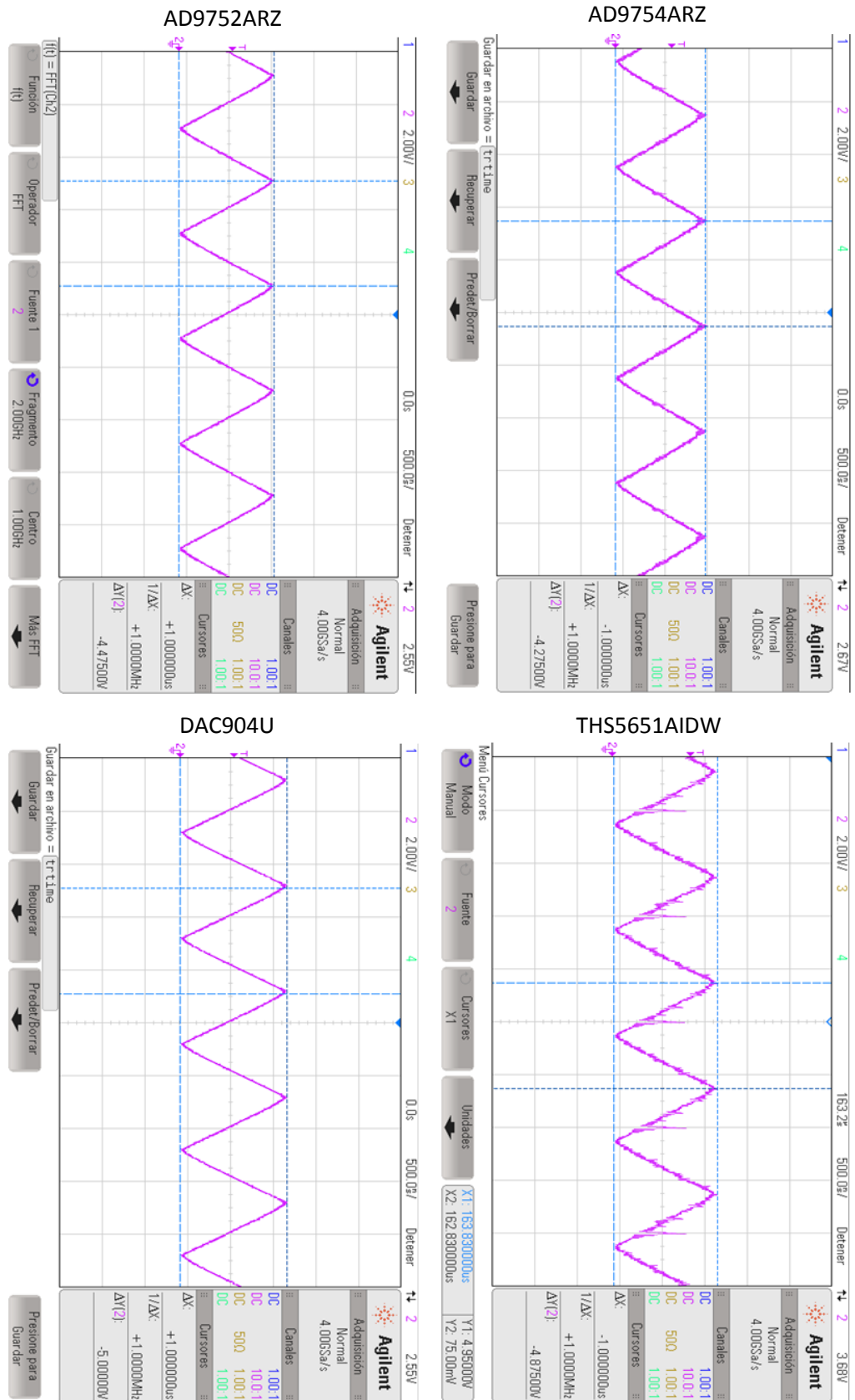


Figura 66: Triangular en el tiempo a 50 kHz

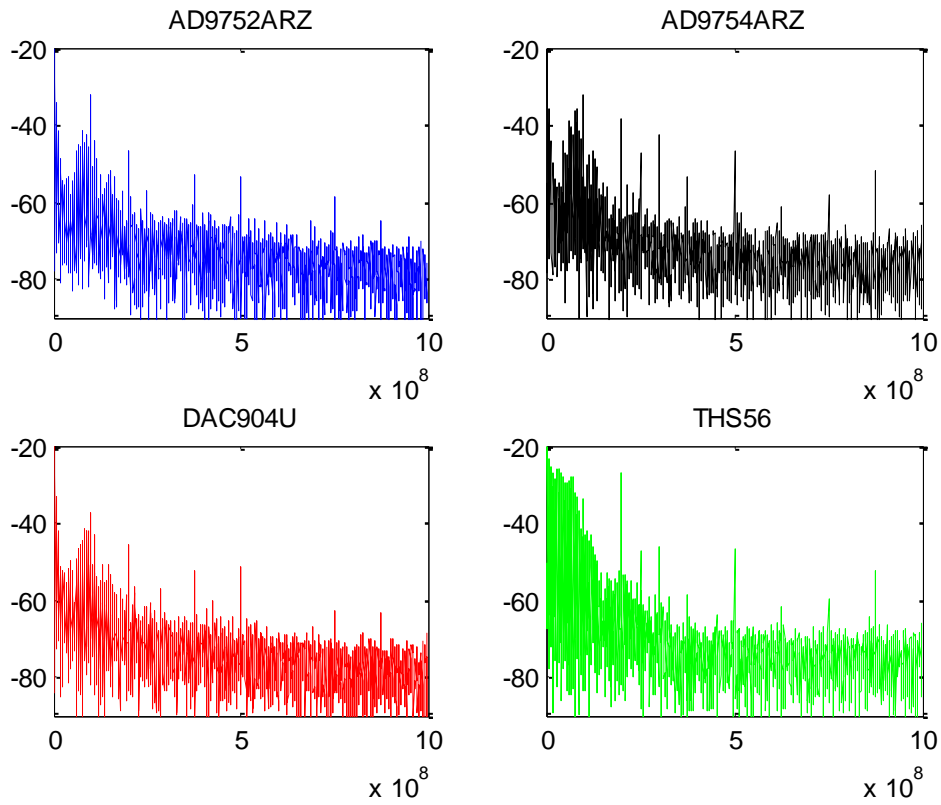


Figura 67: Espectro de la señal triangular de 1 MHz en banda ancha

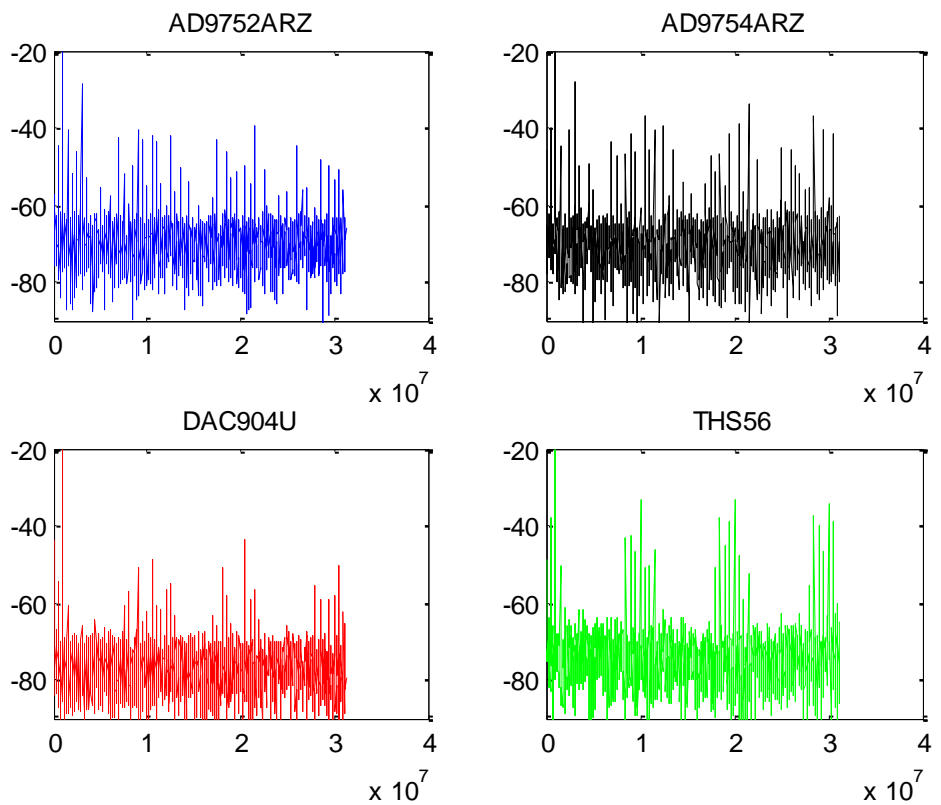


Figura 68: Espectro de la señal triangular de 1 MHz en banda estrecha

ANEXO E Código VHDL de modelos digitalizados

Modelo REAL unifase

```
-----
-- Modelo digitalizado de un conversor Buck unifase
--                               STFG Javier Casatorres 2014-2015
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.MATH_REAL.ALL;

entity Buck is
  port(
    ClkIn  : in std_logic;
    Reset  : in std_logic;
    HSM     : in std_logic;
    LSM     : in std_logic;
    Vin     : in real;
    Iout    : in real;
    Vout    : out real;
    Iin     : out real
  );
end Buck;

architecture Behavioral of Buck is

  constant inct:                real:=20.0e-9;
  constant L:                   real:=22.0e-6;
  constant C:                   real:=220.0e-6;

  constant inct_L :              real:=inct/L;
  constant inct_C :              real:=inct/C;

  signal Iin_INTERNA : real:=0.0;
  signal Vout_INTERNA : real:=0.0;

  signal Ibob:                  real:=0.0;
  signal Icond:                 real:=0.0;
  signal Vbob:                  real:=0.0;

  signal SENALERROR:            std_logic:='0'; --Senal usada para saber si ha
habido un error
begin
  Iin<=Iin_INTERNA;
  Vout<=Vout_INTERNA;
  Icond<=Ibob-Iout;
MULTIPLEXOR: process (HSM,LSM,Ibob,Vout_INTERNA,Iout)
begin
  if HSM='1' and LSM='0' then --Comportamiento de carga
    Vbob<=Vin-Vout_INTERNA;
    Iin_INTERNA<=Ibob;
  elsif HSM='0' and LSM='1' then --Comportamiento de descarga
    Vbob<=-Vout_INTERNA;
    Iin_INTERNA<=0.0;
  elsif HSM='0' and LSM='0' then --Comportamiento residual
    --Entran en juego los diodos acoplados a los mosfet polarizado en
directa
    Iin_INTERNA<=Ibob;
    Vbob<=Vin-Vout_INTERNA;
  else --Corriente de la bobina positiva: diodo LSM polarizado
en directa
    Iin_INTERNA<=0.0;
    Vbob<=-Vout_INTERNA;
  end if;
  else --Caso imposible, ambos MosFet activos: cortocircuito
    SENALERROR<='1';
  end if;
end process MULTIPLEXOR;
```

```

ECDIFFS: process(ClkIn,Reset)
    begin
        if Reset='1' then
            Vout_INTERNA<=0.0; --Al inicio el condensador esta
descargado
            Ibob<=0.0;
        elsif rising_edge(ClkIn) then
            Vout_INTERNA<=Vout_INTERNA+Icond*inct_C;
            Ibob<=Ibob+Vbob*inct_L;
        end if;
    end process;

```

Código 11: Modelo digitalizado en formato REAL del Buck unifase

Testbench del modelo REAL unifase

```
-----
-- Testbench del modelo digitalizado de un conversor Buck unifase
--                                     TFG Javier Casatorres 2014-2015
-----

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
use IEEE.MATH_REAL.ALL;

ENTITY ProbadorBuckLazoAbierto IS
END ProbadorBuckLazoAbierto;

ARCHITECTURE behavior OF ProbadorBuckLazoAbierto IS

    COMPONENT Buck
    PORT (
        ClkIn : IN std_logic;
        Reset : IN std_logic;
        HSM : IN std_logic;
        LSM : IN std_logic;
        Vin : IN real;
        Iout : IN real;
        Vout : OUT real;
        Iin : OUT real
    );
    END COMPONENT;

    signal empezado : std_logic:='0';
--Inputs
    signal ClkIn : std_logic := '0';
    signal Reset : std_logic := '0';
    signal HSM : std_logic := '0';
    signal LSM : std_logic := '0';
    signal VinTB : real := 0.0;
    signal IoutTB : real := 0.0;

--Outputs
    signal VoutTB : real;
    signal IinTB : real;

    -- Clock period definitions
    constant ClkIn_period : time := 20 ns;
    constant Gout: real:= 0.4; -- 2.5 ohm => 2 A
    constant cuentaHasta: integer:= 249; --Contar hasta 249 implica dividir la
frecuencia por 250
    constant TiempoMuerto: integer:=2; --Numero de ciclos de reloj en que no hay activo
ningún MosFet

    --constant dutyCycle: integer:=104; --250*5/12=104 Para conseguir el valor medio
deseado
    --NOTA: Lo normal seria fijar el duty cycle a 5/12 de cuentaHasta+1, pero eso hará
que la
    --tensión de salida sea ligeramente menor de 5 V
    --Habrà que tener en cuenta que el tiempo total sigue siendo 250, pero que el tiempo
a uno
    --no es dutyCycle sino dutyCycle-TiempoMuerto

    constant dutyCycle: integer:= 104+TiempoMuerto;
--Variables
    signal count : integer range 0 to cuentaHasta := 0; --El contador ira hasta 250, de
forma que si
    --el reloj va a 50 MHz, la fconm=200kHz
```

```

BEGIN
    uut: Buck PORT MAP (
        ClkIn => ClkIn,
        Reset => Reset,
        HSM => HSM,
        LSM => LSM,
        Vin => VinTB,
        Iout => IoutTB,
        Vout => VoutTB,
        Iin => IinTB
    );

    ----- Reloj -----
    ClkIn_process :process
    begin
        ClkIn <= '0';
        wait for ClkIn_period/2;
        ClkIn <= '1';
        wait for ClkIn_period/2;
    end process;

    ----- Procesos de PWMs -----

    Count_process :process(ClkIn,Reset)
    begin
        if Reset='1' then
            count<=0;
        elsif rising_edge(ClkIn) then
            if count<249 then
                count<=count+1;
            else
                count<=0;
            end if;
        end if;
    end process;

    HSM_process :process(count)
    begin
        if count <= dutyCycle-TiempoMuerto/2 and count >= TiempoMuerto/2 then
            HSM<='1';
        else
            HSM<='0';
        end if;
    end process;

    LSM_process :process(count)
    begin
        if count > dutyCycle+TiempoMuerto/2 and count < cuentaHasta-TiempoMuerto/2 then
            LSM<='1';
        else
            LSM<='0';
        end if;
    end process;

    -----

    VinTB<=12.0;
    -- Stimulus process
    stim_proc: process
    begin
        Reset<='1';
        -- hold reset state for 100 ns.
        wait for 100 ns;
        Reset<='0';
        wait;
    end process;

    --IoutTB<=0.0;
    iOut_process: process(VoutTB)
    begin
        if empezado='0' then
            empezado<='1';
        else
            IoutTB<=VoutTB*Gout;--VoutTB*Gout;
        end if;
    end process;
END;

```

Código 12: Testbench del modelo REAL unifase

Modelo cuadrifase en coma fija con segmentación

```
-----
-- Modelo en coma fija de un Buck a 50 MHz con segmentación
-- TFG Javier Casatorres Agüero 2014-2015
-----

library IEEE,ieee_proposed, WORK;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use ieee_proposed.fixed_pkg.all; -- ieee_proposed for compatibility version


entity Buck is
  port(
    CLK: in std_logic;
    Reset : in std_logic;

    HSM1 : in std_logic;
    LSM1 : in std_logic;

    HSM2 : in std_logic;
    LSM2 : in std_logic;

    HSM3 : in std_logic;
    LSM3 : in std_logic;

    HSM4 : in std_logic;
    LSM4 : in std_logic;

    --VE: Variable de ESTADO, variables de las ecuaciones en diferencias en que se va
    recalculando el valor en cada iteración.
    --Su precisión es crítica por ser los fallos acumulativos.

    --Dados los ADC y DACs a usar, con una resolución de 10 a 16 bits, hacer que las señales de
    entrada y salida del sistema
    --tengan más resolución no es útil.
    Vin : in STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale 12, así pues Q4.11 -> 15 bits +
    signo
    Iout : in STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale menos de 14, así pues Q4.11 -> 15
    bits + signo

    Vout : out STD_LOGIC_VECTOR(15 downto 0);--NO ES VE. Nunca superará el valor de la entrada, Q4.11 -
    > 15 bits + signo
    Iin : out STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale menos de 20, así pues Q5.10 -> 15
    bits + signo
    --Vout_REAL : out real;
    --Iin_real : out real;

    --Corrientes de las bobinas que serán entradas al regulador
    IL1_out: out STD_LOGIC_VECTOR(15 downto 0);
    IL2_out: out STD_LOGIC_VECTOR(15 downto 0);
    IL3_out: out STD_LOGIC_VECTOR(15 downto 0);
    IL4_out: out STD_LOGIC_VECTOR(15 downto 0)
  );
end Buck;

architecture Behavioral of Buck is

  --Los valores para los que se han calculado inct_L=inct/L; y inct_C=inct/C son los siguientes
  --constant inct: real:=20.0e-9;
  --constant L: real:=22.0e-6;
  --constant C: real:=220.0e-6;

  --El límite superior para los incrementos es aquel que permite que el MSB se ponga a 1, de
  forma que no se desperdicien
  --bits=>floor(log2(valor)). El inferior, el que ocupe 18 bits, que permitirá maximizar la
  precisión a la par que usar la
  --aceleración que proporcionan los multiplicadores de 18 bits que hay embebidos en la FPGA
  usada (Spartan 3)
  constant inct_L : sfixed(-10 downto -27) := to_sfixed(0.000909090,-10,-
  27);--18 bits, Q-10.27
  constant inct_C : sfixed(-13 downto -30) := to_sfixed(0.000090909,-13,-30);--18
  bits, Q-13.30
```

```

--La señal sumIL debería tener dos bits más que las corrientes que se suman por sumarse 4.
Además, se ha determinado
--en la simulación en REAL que el valor máximo es de en torno a 34 A => 6 bits de parte
entera. El límite inferior lo
--determina la resolución de las corrientes en las bobinas (23 bits de parte fraccionaria).
signal sumIL : sfixed(6 downto -23);
--No necesita más de 5 bits para la parte entera. Respetamos la resolución de la suma de
corrientes para la parte decimal.
signal Icond : sfixed(5 downto -23) := (others => '0'); --18 bits

--Si aprovechamos los multiplicadores de 18 bits de la Spartan 3, podremos acelerar ahorrando
área. Así pues, el límite superior
--lo determina el valor máximo de las tensiones en las bobinas para evitar que saturé (9 A), y
el mínimo será el que haga que se
--usen 18 bits.
signal VL1: sfixed(4 downto -13) := (others => '0');
signal VL2: sfixed(4 downto -13) := (others => '0');
signal VL3: sfixed(4 downto -13) := (others => '0');
signal VL4: sfixed(4 downto -13) := (others => '0');

signal VL1_reg: sfixed(4 downto -13) := (others => '0');
signal VL2_reg: sfixed(4 downto -13) := (others => '0');
signal VL3_reg: sfixed(4 downto -13) := (others => '0');
signal VL4_reg: sfixed(4 downto -13) := (others => '0');

signal SENALERROR: std_logic:='0'; --Senal usada para saber si ha habido un error,
pues indica la entrada en un estado prohibido

--Variables de estado

signal Vout_INTERNA : sfixed(4 downto -23) := (others => '0');
signal IncVout : sfixed(4 downto -23) := (others => '0');

--Sumamos el valor anterior a algo con formato Q-14.29 * Q5.12 => Q-8.41 No es factible tanta
precisión,
--porque tenemos que sumarlo a algo que va a llegar hasta valores de Vin V (4 bits de parte
entera).
--Se usa en cambio el siguiente razonamiento:
--  $Vo(k) = Vo(k-1) * (inct/C) * Icond$  => Tomamos Icond para el calculo como 5% del valor máximo en
régimen permanente (0.0752 V). Multiplicamos por (inct/C),
-- y hallamos el logaritmo en base 2 de eso. Al resultado le sumamos 8 bits para obtener una
precisión suficiente. Eso indica 23
-- bits para la parte fraccionaria

signal IL1: sfixed(4 downto -18) := (others => '0');
signal IL2: sfixed(4 downto -18) := (others => '0');
signal IL3: sfixed(4 downto -18) := (others => '0');
signal IL4: sfixed(4 downto -18) := (others => '0');

signal IncIL1: sfixed(4 downto -18) := (others => '0');
signal IncIL2: sfixed(4 downto -18) := (others => '0');
signal IncIL3: sfixed(4 downto -18) := (others => '0');
signal IncIL4: sfixed(4 downto -18) := (others => '0');
--Para las corrientes el razonamiento es análogo al anterior: no podemos usar toda la
resolución necesaria para evitar el error.
--Tomamos IL(k)=IL(k-1)*(inct/L)*VL(k), y sabemos que VL llega al valor máximo de 7 V en
régimen permanente.
--Por tanto, 5% de 12 * inct/L => -10 para el límite superior + 8 bits para la resolución =>
18 bits de parte fraccionaria. El límite
--superior lo determina el valor máximo: casi 9 A en todos los casos (determinados por la
simulación en REAL), es decir, 4 bits.

--Las corrientes de entrada a cada etapa pueden ser o bien 0 o bien la corriente que circule
por la bobina de esa etapa, así que las
--señales deberán tener la misma resolución que las corrientes de las bobinas.
signal Iin_etapa1: sfixed(4 downto -18) := (others => '0');
signal Iin_etapa2: sfixed(4 downto -18) := (others => '0');
signal Iin_etapa3: sfixed(4 downto -18) := (others => '0');
signal Iin_etapa4: sfixed(4 downto -18) := (others => '0');

begin
--Registros del Pipeline
process(CLK,Reset)
begin

```

```

    if Reset='1' then
        SumIL<=(others=>'0');
        Icond<=(others=>'0');
    elsif rising_edge(CLK) then
        SumIL<=resize(IL1+IL2+IL3+IL4,SumIL);
        Icond<=resize(SumIL-to_sfixed(Iout,4,-11),Icond);
    end if;
end process;

```

```

--Asignacion a señales de salida
Iin<= to_slv(resize(Iin_etapa1+Iin_etapa2+Iin_etapa3+Iin_etapa4,5,-10));
Vout<= to_slv(resize(Vout_INTERNA,4,-11));

IL1_out <= to_slv(resize(IL1,4,-11));
IL2_out <= to_slv(resize(IL2,4,-11));
IL3_out <= to_slv(resize(IL3,4,-11));
IL4_out <= to_slv(resize(IL4,4,-11));

--Vout REAL <= to_real(Vout INTERNA);
--Iin_REAL <= to_real(Iin_etapa1+Iin_etapa2+Iin_etapa3+Iin_etapa4);

```

MULTIPLEXOR:

```

process (HSM1, LSM1, HSM2, LSM2, HSM3, LSM3, HSM4, LSM4, IL1, IL2, IL3, IL4, Vout_INTERNA, Iout, Vin, VL1, Iin_etapa1, VL2, Iin_etapa2, VL3, Iin_etapa3, VL4, Iin_etapa4)
begin
    -----BOBINA 1-----
    if HSM1='1' and LSM1='0' then
        VL1<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL1);
        Iin_etapa1<=IL1;
    elsif HSM1='0' and LSM1='1' then
        VL1<=resize(-Vout_interna,VL1);
        Iin_etapa1<=to_sfixed(0.0,Iin_etapa1);
    elsif HSM1='0' and LSM1='0' then --Tiempo muerto
        if IL1>=to_sfixed(0.0,IL1) then
            VL1<=resize(-Vout_interna,VL1);
            Iin_etapa1<=to_sfixed(0.0,Iin_etapa1);
        else
            VL1<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL1);
            Iin_etapa1<=IL1;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL1<=(others=>'0');
        Iin_etapa1<=(others=>'0');
    end if;

    -----BOBINA 2-----
    if HSM2='1' and LSM2='0' then
        VL2<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL2);
        Iin_etapa2<=IL2;
    elsif HSM2='0' and LSM2='1' then
        VL2<=resize(-Vout_interna,VL2);
        Iin_etapa2<=to_sfixed(0.0,Iin_etapa2);
    elsif HSM2='0' and LSM2='0' then --Tiempo muerto
        if IL2>=to_sfixed(0.0,IL2) then
            VL2<=resize(-Vout_interna,VL2);
            Iin_etapa2<=to_sfixed(0.0,Iin_etapa2);
        else
            VL2<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL2);
            Iin_etapa2<=IL2;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL2<=(others=>'0');
        Iin_etapa2<=(others=>'0');
    end if;

    -----BOBINA 3-----
    if HSM3='1' and LSM3='0' then
        VL3<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL3);
        Iin_etapa3<=IL3;
    elsif HSM3='0' and LSM3='1' then
        VL3<=resize(-Vout_interna,VL3);
        Iin_etapa3<=to_sfixed(0.0,Iin_etapa3);
    elsif HSM3='0' and LSM3='0' then --Tiempo muerto
        if IL3>=0.0 then
            VL3<=resize(-Vout_interna,VL3);

```

```

        Iin_etapa3<=to_sfised(0.0,Iin_etapa3);
    else
        VL3<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL3);
        Iin_etapa3<=IL3;
    end if;
else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
    SENALERROR <= '1';
    VL3<=(others=>'0');
    Iin_etapa3<=(others=>'0');
end if;
-----BOBINA 4-----
    if HSM4='1' and LSM4='0' then
        VL4<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL4);
        Iin_etapa4<=IL4;
    elsif HSM4='0' and LSM4='1' then
        VL4<=resize(-Vout_interna,VL4);
        Iin_etapa4<=to_sfised(0.0,Iin_etapa4);
    elsif HSM4='0' and LSM4='0' then --Tiempo muerto
        if IL4>=0.0 then
            VL4<=resize(-Vout_interna,VL4);
            Iin_etapa4<=to_sfised(0.0,Iin_etapa4);
        else
            VL4<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL4);
            Iin_etapa4<=IL4;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL4<=(others=>'0');
        Iin_etapa4<=(others=>'0');
    end if;
end process MULTIPLEXOR;

process (CLK,Reset)
begin
    if Reset='1' then
        VL1_reg<=(others=>'0');
        VL2_reg<=(others=>'0');
        VL3_reg<=(others=>'0');
        VL4_reg<=(others=>'0');
    elsif rising_edge(CLK) then
        VL1_reg<=VL1;
        VL2_reg<=VL2;
        VL3_reg<=VL3;
        VL4_reg<=VL4;
    end if;
end process;

--Proceso para modelar el desarrollo de las ecuaciones en diferencias
ECDIFFS: process(CLK,Reset)
begin
    if Reset='1' then
        Vout_INTERNA<=to_sfised(0.0,Vout_INTERNA); --Al inicio el condensador esta
descargado

        --Incluimos una etapa de pipeline adicional para entrar en tiempos
        IncIL1<=to_sfised(0.0,IncIL1);
        IncIL2<=to_sfised(0.0,IncIL2);
        IncIL3<=to_sfised(0.0,IncIL3);
        IncIL4<=to_sfised(0.0,IncIL4);

        IL1<=to_sfised(0.0,IL1);
        IL2<=to_sfised(0.0,IL2);
        IL3<=to_sfised(0.0,IL3);
        IL4<=to_sfised(0.0,IL4);
    elsif CLK'event and Clk='1' then
        IncVout<=resize(Icond*inct_C,IncVout);
        Vout_INTERNA<=resize(Vout_INTERNA+IncVout,Vout_INTERNA);

        IncIL1<=resize(VL1_reg*inct_L,IncIL1);
        IncIL2<=resize(VL2_reg*inct_L,IncIL2);
        IncIL3<=resize(VL3_reg*inct_L,IncIL3);
        IncIL4<=resize(VL4_reg*inct_L,IncIL4);

        IL1<=resize(IL1+IncIL1,IL1);
        IL2<=resize(IL2+IncIL2,IL2);
        IL3<=resize(IL3+IncIL3,IL3);
        IL4<=resize(IL4+IncIL4,IL4);
    end if;
end process;

```

```

end process ECDIFFS;

end Behavioral;

```

Código 13: Modelo en coma fija con segmentación a 50 MHz

Generador de PWMs desfasados con reloj de 50 MHz

```

-----
-- Generador de PWM para frecuencia de 50 MHz
-- TFG Javier Casatorres Agüero 2014-2015
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity PWMGenerator is
    Port ( HSM1 : out  STD_LOGIC;
          HSM2 : out  STD_LOGIC;
          HSM3 : out  STD_LOGIC;
          HSM4 : out  STD_LOGIC;
          LSM1 : out  STD_LOGIC;
          LSM2 : out  STD_LOGIC;
          LSM3 : out  STD_LOGIC;
          LSM4 : out  STD_LOGIC;
          Reset : in   STD_LOGIC;
          CLK   : in   STD_LOGIC);
end PWMGenerator;

architecture Behavioral of PWMGenerator is

    constant cuentaHasta: integer:= 249; --Contar hasta 249 implica dividir la frecuencia por 250
    constant TiempoMuerto: integer:=2; --Numero de ciclos de reloj en que no hay activo ningún
MosFet

    --constant dutyCycle: integer:=104; --250*5/12=104 Para conseguir el valor medio deseado
    --NOTA: Lo normal seria fijar el duty cycle a 5/12 de cuentaHasta+1, pero eso hará que la
tensión de salida sea ligeramente menor de 5 V
    --Habrà que tener en cuenta que el tiempo total sigue siendo 250, pero que el tiempo a uno no
es dutyCycle sino dutyCycle-TiempoMuerto

    constant dutyCycle: integer:= 104+TiempoMuerto;

    --Contador principal para PWM
    signal count : integer range 0 to cuentaHasta := 0; --El contador ira hasta 250, de forma que
si el reloj va a 50 MHz, la fconm=200kHz
    --Contadores desfasados
    signal count2 : integer range 0 to cuentaHasta := 0; --Desfasado 90°
    signal count3 : integer range 0 to cuentaHasta := 0; --Desfasado 180°
    signal count4 : integer range 0 to cuentaHasta := 0; --Desfasado 270°

    --Señales para poder registrar los ciclos PWM para realizar el pipeline y cumplir tiempos
    signal HSM1_aux : STD_LOGIC;
    signal HSM2_aux : STD_LOGIC;
    signal HSM3_aux : STD_LOGIC;
    signal HSM4_aux : STD_LOGIC;

    signal LSM1_aux : STD_LOGIC;
    signal LSM2_aux : STD_LOGIC;
    signal LSM3_aux : STD_LOGIC;
    signal LSM4_aux : STD_LOGIC;

begin
    Count_process : process (CLK, Reset)
    begin
        if Reset='1' then
            count<=0;
            count2<=(cuentaHasta+1)/4;
            count3<=2*(cuentaHasta+1)/4;
            count4<=3*(cuentaHasta+1)/4;
        elsif rising_edge(CLK) then
            if count<249 then

```

```

        count<=count+1;
    else
        count<=0;
    end if;

    if count2<249 then
        count2<=count2+1;
    else
        count2<=0;
    end if;

    if count3<249 then
        count3<=count3+1;
    else
        count3<=0;
    end if;

    if count4<249 then
        count4<=count4+1;
    else
        count4<=0;
    end if;
end if;
end process;
--PWM1
HSM1_process :process(count)
begin
    if count <= dutyCycle-TiempoMuerto/2 and count >= TiempoMuerto/2 then
        HSM1_aux<='1';
    else
        HSM1_aux<='0';
    end if;
end process;
LSM1_process :process(count)
begin
    if count > dutyCycle+TiempoMuerto/2 and count < cuentaHasta-TiempoMuerto/2 then
        LSM1_aux<='1';
    else
        LSM1_aux<='0';
    end if;
end process;

--PWM2
HSM2_process :process(count2)
begin
    if count2 <= dutyCycle-TiempoMuerto/2 and count2 >= TiempoMuerto/2 then
        HSM2_aux<='1';
    else
        HSM2_aux<='0';
    end if;
end process;
LSM2_process :process(count2)
begin
    if count2 > dutyCycle+TiempoMuerto/2 and count2 < cuentaHasta-TiempoMuerto/2 then
        LSM2_aux<='1';
    else
        LSM2_aux<='0';
    end if;
end process;

--PWM3
HSM3_process :process(count3)
begin
    if count3 <= dutyCycle-TiempoMuerto/2 and count3 >= TiempoMuerto/2 then
        HSM3_aux<='1';
    else
        HSM3_aux<='0';
    end if;
end process;
LSM3_process :process(count3)
begin
    if count3 > dutyCycle+TiempoMuerto/2 and count3 < cuentaHasta-TiempoMuerto/2 then
        LSM3_aux<='1';
    else
        LSM3_aux<='0';
    end if;
end process;

```

```

--PWM4
HSM4_process :process(count4)
begin
    if count4 <= dutyCycle-TiempoMuerto/2 and count4 >= TiempoMuerto/2 then
        HSM4_aux<='1';
    else
        HSM4_aux<='0';
    end if;
end process;
LSM4_process :process(count4)
begin
    if count4 > dutyCycle+TiempoMuerto/2 and count4 < cuentaHasta-TiempoMuerto/2 then
        LSM4_aux<='1';
    else
        LSM4_aux<='0';
    end if;
end process;

--Proceso de registro de las señales de salida
process(CLK, Reset)
begin
    if Reset='1' then
        HSM1<='0';
        HSM2<='0';
        HSM3<='0';
        HSM4<='0';
        LSM1<='0';
        LSM2<='0';
        LSM3<='0';
        LSM4<='0';
    elsif rising_edge(CLK) then
        HSM1<=HSM1_aux;
        HSM2<=HSM2_aux;
        HSM3<=HSM3_aux;
        HSM4<=HSM4_aux;
        LSM1<=LSM1_aux;
        LSM2<=LSM2_aux;
        LSM3<=LSM3_aux;
        LSM4<=LSM4_aux;
    end if;
end process;
end Behavioral;

```

Código 14: Generador de PWMs con reloj a 50 MHz

Modelo cuadrifase en coma fija a 20 MHz

```

-----
-- Modelo digital de un Buck en coma fija a 20 MHz
-- TFG Javier Casatorres Agüero 2014-2015
-----

library IEEE,ieee_proposed, WORK;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use ieee_proposed.fixed_pkg.all; -- ieee_proposed for compatibility version

entity Buck is
    port(
        CLK: in std_logic;
        Reset : in std_logic;

        HSM1 : in std_logic;
        LSM1 : in std_logic;

        HSM2 : in std_logic;
        LSM2 : in std_logic;

        HSM3 : in std_logic;
        LSM3 : in std_logic;
    );
end entity;

```

```

HSM4    : in std_logic;
LSM4    : in std_logic;
--VE: Variable de ESTADO, variables de las ecuaciones en diferencias en que se va
recalculando el valor en cada iteración. Su precisión es crítica por ser los fallos acumulativos.

--Dados los ADC y DACs a usar, con una resolución de 10 a 16 bits, hacer que las señales de
entrada y salida del sistema tengan más resolución no es útil.
Vin      : in STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale 12, así pues Q4.11 -> 15 bits +
signo
Iout     : in STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale menos de 14, así pues Q4.11 -> 15
bits + signo
Vout     : out STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Nunca superará el valor de la
entrada, Q4.11 -> 15 bits + signo
Iin      : out STD_LOGIC_VECTOR(15 downto 0); --NO ES VE. Vale menos de 20, así pues Q5.10 ->
15 bits + signo

--Corrientes de las bobinas que serán entradas al regulador
IL1_out : out STD_LOGIC_VECTOR(15 downto 0);
IL2_out : out STD_LOGIC_VECTOR(15 downto 0);
IL3_out : out STD_LOGIC_VECTOR(15 downto 0);
IL4_out : out STD_LOGIC_VECTOR(15 downto 0);
);
end Buck;

architecture Behavioral of Buck is

--Los valores para los que se han calculado inct L=inct/L; y inct_C=inct/C son los siguientes
--constant inct:                      real:=50.0e-9;
--constant L:                        real:=22.0e-6;
--constant C:                        real:=220.0e-6;

--El límite superior para los incrementos es aquel que permite que el MSB se ponga a 1, de
forma que no se desperdicien
--bits=>floor(log2(valor)). El inferior, el que ocupe 18 bits, que permitirá maximizar la
precisión a la par que usar la
--aceleración que proporcionan los multiplicadores de 18 bits que hay embebidos en la FPGA
usada (Spartan 3)

constant inct_L :                      sfixed(-8 downto -25) := to_sfixed(0.002272727,-8,-25);-
-18 bits, Q-9.26
constant inct_C :                      sfixed(-9 downto -26) := to_sfixed(0.0002272727,-12,-29);--18
bits, Q-13.30

--La señal sumIL debería tener dos bits más que las corrientes que se suman por sumarse 4.
Además, se ha determinado en la simulación en REAL que el valor máximo es de en torno a 34 A => 6
bits de parte entera. El límite inferior lo determina la resolución de las corrientes en las
bobinas (23 bits de parte fraccionaria).
signal sumIL :                      sfixed(6 downto -23);
--No necesita más de 5 bits para la parte entera. Respetamos la resolución de la suma de
corrientes para la parte decimal.
signal Icond :                      sfixed(5 downto -23) := (others => '0'); --18 bits

--Si aprovechamos los multiplicadores de 18 bits de la Spartan 3, podremos acelerar ahorrando
área. Así pues, el límite superior lo determina el valor máximo de las tensiones en las bobinas
para evitar que sature (9 A), y el mínimo será el que haga que se usen 18 bits.
signal VL1:                          sfixed(4 downto -13) := (others => '0');
signal VL2:                          sfixed(4 downto -13) := (others => '0');
signal VL3:                          sfixed(4 downto -13) := (others => '0');
signal VL4:                          sfixed(4 downto -13) := (others => '0');

signal SENALERRO:                    std_logic:='0'; --Senal usada para saber si ha habido un error,
pues indica la entrada en un estado prohibido

--Variables de estado

signal Vout_INTERNA : sfixed(4 downto -23) := (others => '0');
--Sumamos el valor anterior a algo con formato Q-14.29 * Q5.12 => Q-8.41 No es factible tanta
precision,
--porque tenemos que sumarlo a algo que va a llegar hasta valores de Vin V (4 bits de parte
entera).
--Se usa en cambio el siguiente razonamiento:
-- Vo(k)=Vo(k-1)*(inct/C)*Icond => Tomamos Icond para el calculo como 5% del valor máximo en
régimen permanente(0.0752 V). Multiplicamos por (inct/C),
-- y hallamos el logaritmo en base 2 de eso. Al resultado le sumamos 8 bits para obtener una
precisión suficiente. Eso indica 23
-- bits para la parte fraccionaria

```



```

signal IL1:                                sfixed(4 downto -18) := (others => '0');
signal IL2:                                sfixed(4 downto -18) := (others => '0');
signal IL3:                                sfixed(4 downto -18) := (others => '0');
signal IL4:                                sfixed(4 downto -18) := (others => '0');
--Para las corrientes el razonamiento es análogo al anterior: no podemos usar toda la
resolución necesaria para evitar el error.
--Tomamos IL(k)=IL(k-1)*(inct/L)*VL(k), y sabemos que VL llega al valor máximo de 7 V en
régimen permanente.
--Por tanto, 5% de 12 * inct/L => -10 para el límite superior + 8 bits para la resolución =>
18 bits de parte fraccionaria. El límite
--superior lo determina el valor máximo: casi 9 A en todos los casos (determinados por la
simulación en REAL), es decir, 4 bits.

--Las corrientes de entrada a cada etapa pueden ser o bien 0 o bien la corriente que circule
por la bobina de esa etapa, así que las
--señales deberán tener la misma resolución que las corrientes de las bobinas.
signal Iin_etapa1:                          sfixed(4 downto -18) := (others => '0');
signal Iin_etapa2:                          sfixed(4 downto -18) := (others => '0');
signal Iin_etapa3:                          sfixed(4 downto -18) := (others => '0');
signal Iin_etapa4:                          sfixed(4 downto -18) := (others => '0');

begin

SumIL<=resize(IL1+IL2+IL3+IL4,SumIL);
Icond<=resize(SumIL-to_sfixed(Iout,4,-11),Icond);

--Asignación a señales de salida
process (CLK,Reset)
begin
    if Reset='1' then
        Iin<= (others=>'0');
        Vout<= (others=>'0');
        IL1_out <= (others=>'0');
        IL2_out <= (others=>'0');
        IL3_out <= (others=>'0');
        IL4_out <= (others=>'0');
    elsif rising_edge(CLK) then
        Iin<= to_slv(resize(Iin_etapa1+Iin_etapa2+Iin_etapa3+Iin_etapa4,5,-10));
        Vout<= to_slv(resize(Vout_INTERNA,4,-11));
        IL1_out <= to_slv(resize(IL1,4,-11));
        IL2_out <= to_slv(resize(IL2,4,-11));
        IL3_out <= to_slv(resize(IL3,4,-11));
        IL4_out <= to_slv(resize(IL4,4,-11));
    end if;
end process;

MULTIPLEXOR:
process (HSM1,LSM1,HSM2,LSM2,HSM3,LSM3,HSM4,LSM4,IL1,IL2,IL3,IL4,Vout_INTERNA,Iout,Vin,VL1,Iin_etap
a1,VL2,Iin_etapa2,VL3,Iin_etapa3,VL4,Iin_etapa4)
begin
    -----BOBINA 1-----
    if HSM1='1' and LSM1='0' then
        VL1<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL1);
        Iin_etapa1<=IL1;
    elsif HSM1='0' and LSM1='1' then
        VL1<=resize(-Vout_interna,VL1);
        Iin_etapa1<=to_sfixed(0.0,Iin_etapa1);
    elsif HSM1='0' and LSM1='0' then --Tiempo muerto
        if IL1>=to_sfixed(0.0,IL1) then
            VL1<=resize(-Vout_interna,VL1);
            Iin_etapa1<=to_sfixed(0.0,Iin_etapa1);
        else
            VL1<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL1);
            Iin_etapa1<=IL1;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL1<= (others=>'0');
        Iin_etapa1<= (others=>'0');
    end if;
    -----BOBINA 2-----
    if HSM2='1' and LSM2='0' then
        VL2<=resize(to_sfixed(Vin,4,-11)-Vout_interna,VL2);
        Iin_etapa2<=IL2;

```

```

    elsif HSM2='0' and LSM2='1' then
        VL2<=resize(-Vout_interna,VL2);
        Iin_etapa2<=to_sfised(0.0,Iin_etapa2);
    elsif HSM2='0' and LSM2='0' then --Tiempo muerto
        if IL2>=to_sfised(0.0,IL2) then
            VL2<=resize(-Vout_interna,VL2);
            Iin_etapa2<=to_sfised(0.0,Iin_etapa2);
        else
            VL2<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL2);
            Iin_etapa2<=IL2;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL2<=(others=>'0');
        Iin_etapa2<=(others=>'0');
    end if;
    -----BOBINA 3-----
    if HSM3='1' and LSM3='0' then
        VL3<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL3);
        Iin_etapa3<=IL3;
    elsif HSM3='0' and LSM3='1' then
        VL3<=resize(-Vout_interna,VL3);
        Iin_etapa3<=to_sfised(0.0,Iin_etapa3);
    elsif HSM3='0' and LSM3='0' then --Tiempo muerto
        if IL3>=0.0 then
            VL3<=resize(-Vout_interna,VL3);
            Iin_etapa3<=to_sfised(0.0,Iin_etapa3);
        else
            VL3<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL3);
            Iin_etapa3<=IL3;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL3<=(others=>'0');
        Iin_etapa3<=(others=>'0');
    end if;
    -----BOBINA 4-----
    if HSM4='1' and LSM4='0' then
        VL4<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL4);
        Iin_etapa4<=IL4;
    elsif HSM4='0' and LSM4='1' then
        VL4<=resize(-Vout_interna,VL4);
        Iin_etapa4<=to_sfised(0.0,Iin_etapa4);
    elsif HSM4='0' and LSM4='0' then --Tiempo muerto
        if IL4>=0.0 then
            VL4<=resize(-Vout_interna,VL4);
            Iin_etapa4<=to_sfised(0.0,Iin_etapa4);
        else
            VL4<=resize(to_sfised(Vin,4,-11)-Vout_interna,VL4);
            Iin_etapa4<=IL4;
        end if;
    else --Ambos MosFet activos, se debe a un error y hay un cortocircuito
        SENALERROR <= '1';
        VL4<=(others=>'0');
        Iin_etapa4<=(others=>'0');
    end if;
end process MULTIPLEXOR;

--Proceso para modelar el desarrollo de las ecuaciones en diferencias
ECDIFFS: process(CLK,Reset)
begin
    if Reset='1' then
        Vout_INTERNA<=to_sfised(0.0,Vout_INTERNA); --Al inicio el condensador esta
descargado
        IL1<=to_sfised(0.0,IL1);
        IL2<=to_sfised(0.0,IL2);
        IL3<=to_sfised(0.0,IL3);
        IL4<=to_sfised(0.0,IL4);
    elsif CLK'event and Clk='1' then
        Vout_INTERNA<=resize(Vout_INTERNA+Icond*inct_C,Vout_INTERNA);
        IL1<=resize(IL1+VL1*inct_L,IL1);
        IL2<=resize(IL2+VL2*inct_L,IL2);
        IL3<=resize(IL3+VL3*inct_L,IL3);
        IL4<=resize(IL4+VL4*inct_L,IL4);
    end if;
end process ECDIFFS;

```

```
end process ECDIFFS;
end Behavioral;
```

Código 15: Modelo cuadrifase en coma fija a 20 MHz

Modelo de alimentación y carga variable

```
-----
-- Generación de alimentaciones para un modelo BUCK
-- TFG Javier Casatorres Agüero
-- 2014-2015
-----

library IEEE, ieee_proposed,WORK;
use IEEE.STD_LOGIC_1164.ALL;
use ieee_proposed.fixed_pkg.all; -- ieee_proposed for compatibility version

entity AlimentacionBuck is
    Port ( Reset: in std_logic;
          CLK: in std_logic;
          VinBuck : out  STD_LOGIC_VECTOR (15 downto 0);
          IoutBuck : out  STD_LOGIC_VECTOR (15 downto 0);
          IinBuck  : in   STD_LOGIC_VECTOR (15 downto 0);
          VoutBuck : in   STD_LOGIC_VECTOR (15 downto 0);
          Switch1: in   STD_LOGIC;
          Switch2: in   STD_LOGIC);
end AlimentacionBuck;

architecture Behavioral of AlimentacionBuck is
    signal Gout: sfixed(4 downto -11):= to_sfised(1.6,4,-11); --La carga sera de 1 ohmio
begin
    VinBuck<=to_slv(to_sfised(12,4,-11)); --Numero 12 en Q4.11, fijo
    Gout <= to_sfised(1.6,4,-11) when Switch1='1' and Switch2='1' else --Corriente 8 A
        to_sfised(1,4,-11)   when Switch1='1' and Switch2='0' else --Corriente 5 A
        to_sfised(0.5,4,-11) when Switch1='0' and Switch2='1' else --Corriente 2.5 A
        to_sfised(0.2,4,-11);    --Corriente 1 A

    process (CLK,Reset)
    begin
        if Reset='1' then
            IoutBuck<=(others=>'0');
        elsif rising_edge(CLK) then
            IoutBuck<=to_slv(resize(to_sfised(VoutBuck,4,-11)*Gout,4,-11));
        end if;
    end process;
end Behavioral;
```

Código 16: Modelo de alimentación y carga variable para el Buck

ANEXO F Simulaciones del transitorio de V_{out}

Simulación analógica del modelo unifase

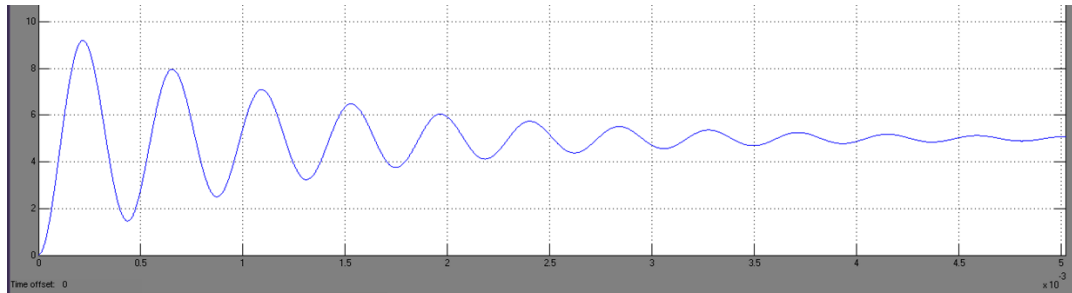


Figura 69: Transitorio de tensión de salida de la simulación analógica unifase

Modelo REAL unifase a 50 MHz

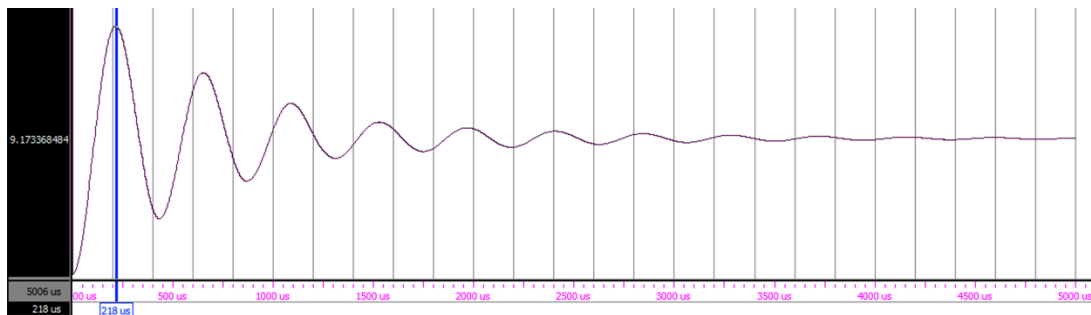


Figura 70: Transitorio de tensión de salida del modelo REAL unifase

Simulación analógica del modelo cuadrifase

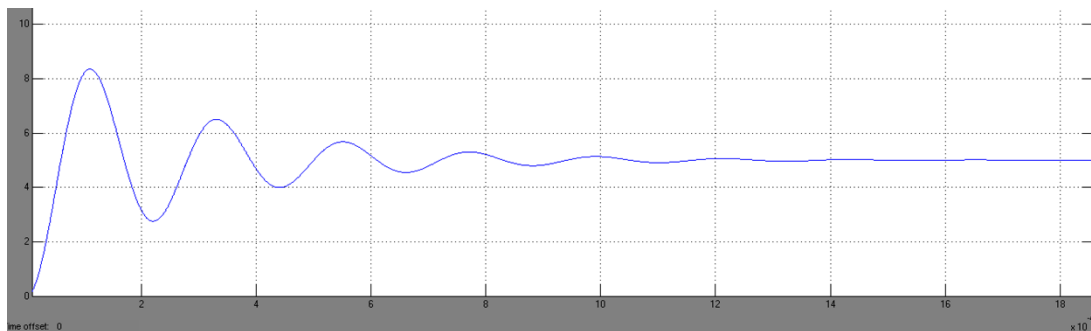


Figura 71: Transitorio de tensión de salida de la simulación analógica cuadrifase

Modelo REAL cuadrifase a 50 MHz

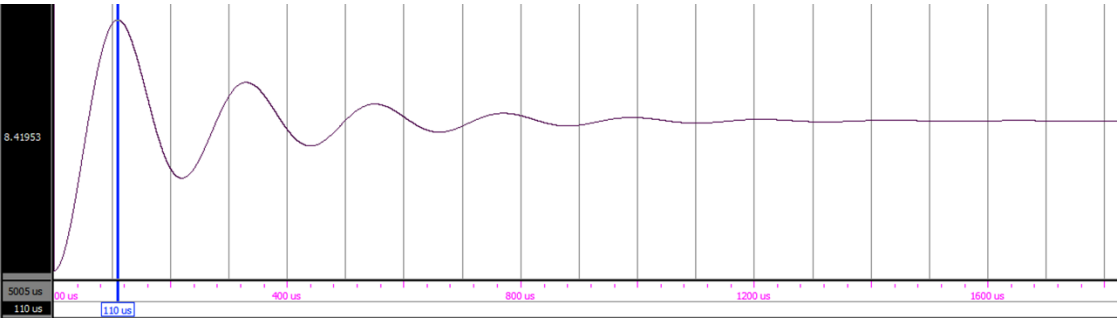


Figura 72: Transitorio de tensión de salida del modelo REAL cuadrifase

Modelo cuadrifase a 50 MHz en coma fija

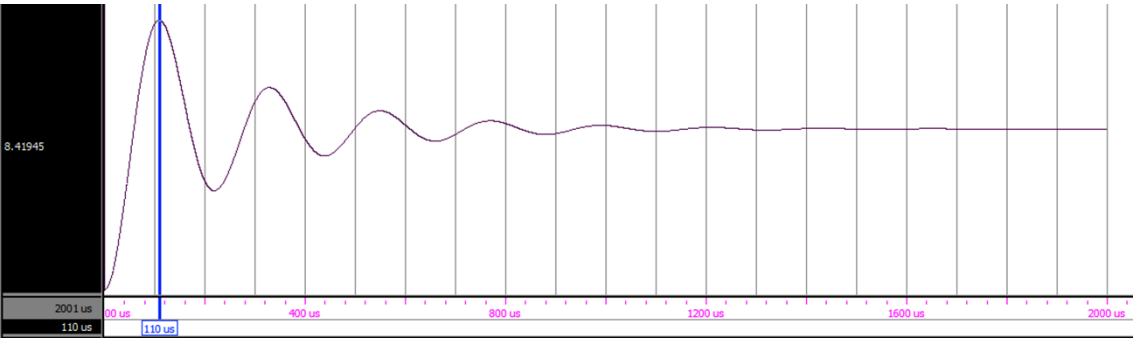


Figura 73: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 50 MHz

Modelo cuadrifase a 50 MHz en coma fija con segmentación

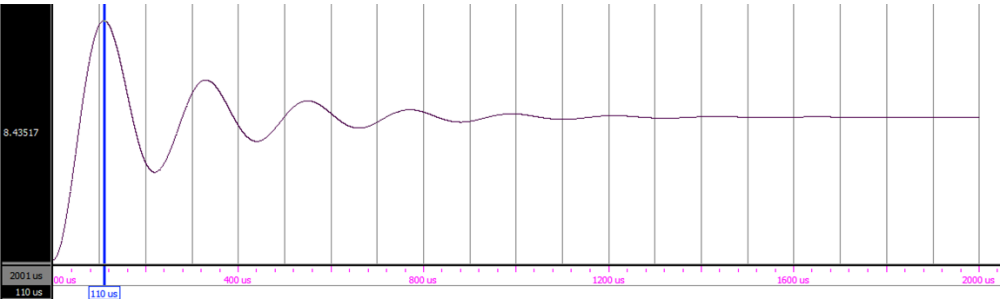


Figura 74: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 50 MHz con segmentación

Modelo cuadrifase a 20 MHz en coma fija

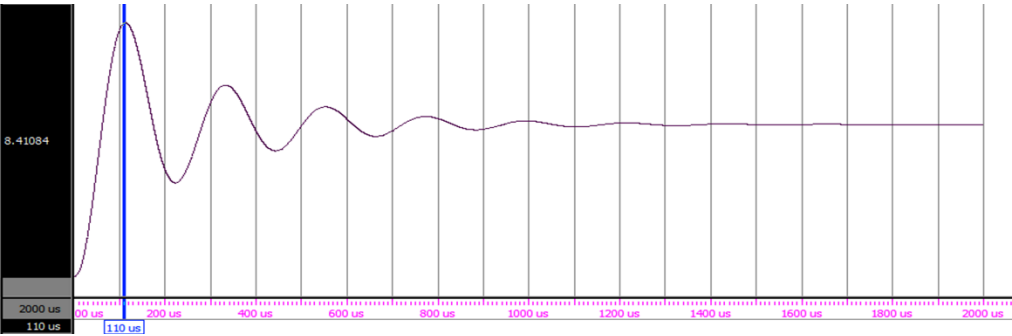


Figura 75: Transitorio de tensión de salida del modelo cuadrifase en coma fija a 20 MHz

ANEXO G Comparativa modelo HIL – realidad

La imagen superior en todos los casos se refiere al modelo HIL del convertidor, mientras que la inferior muestra los resultados del Buck real. Se utiliza la siguiente leyenda:

Modelo HIL	Buck real
<div><div></div> Tensión de salida</div>	<div><div></div> Tensión de salida</div>
<div><div></div> Corriente de fase dos</div>	<div><div></div> Corriente de fase dos</div>
<div><div></div> Corriente de fase tres</div>	<div><div></div> Corriente de fase tres</div>

Corriente de salida de 1 A

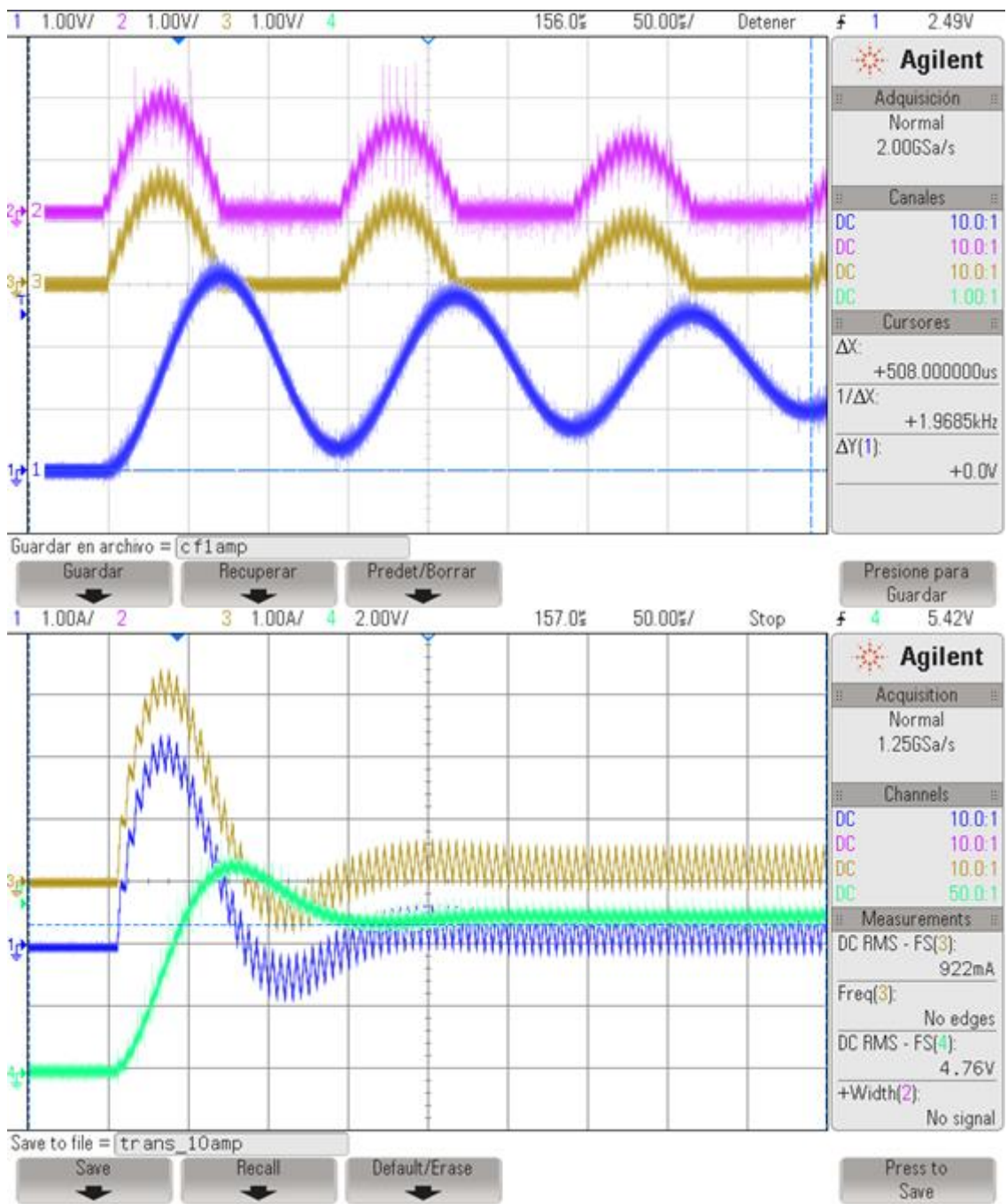


Figura 76: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 1 A

Corriente de salida de 2,5 A

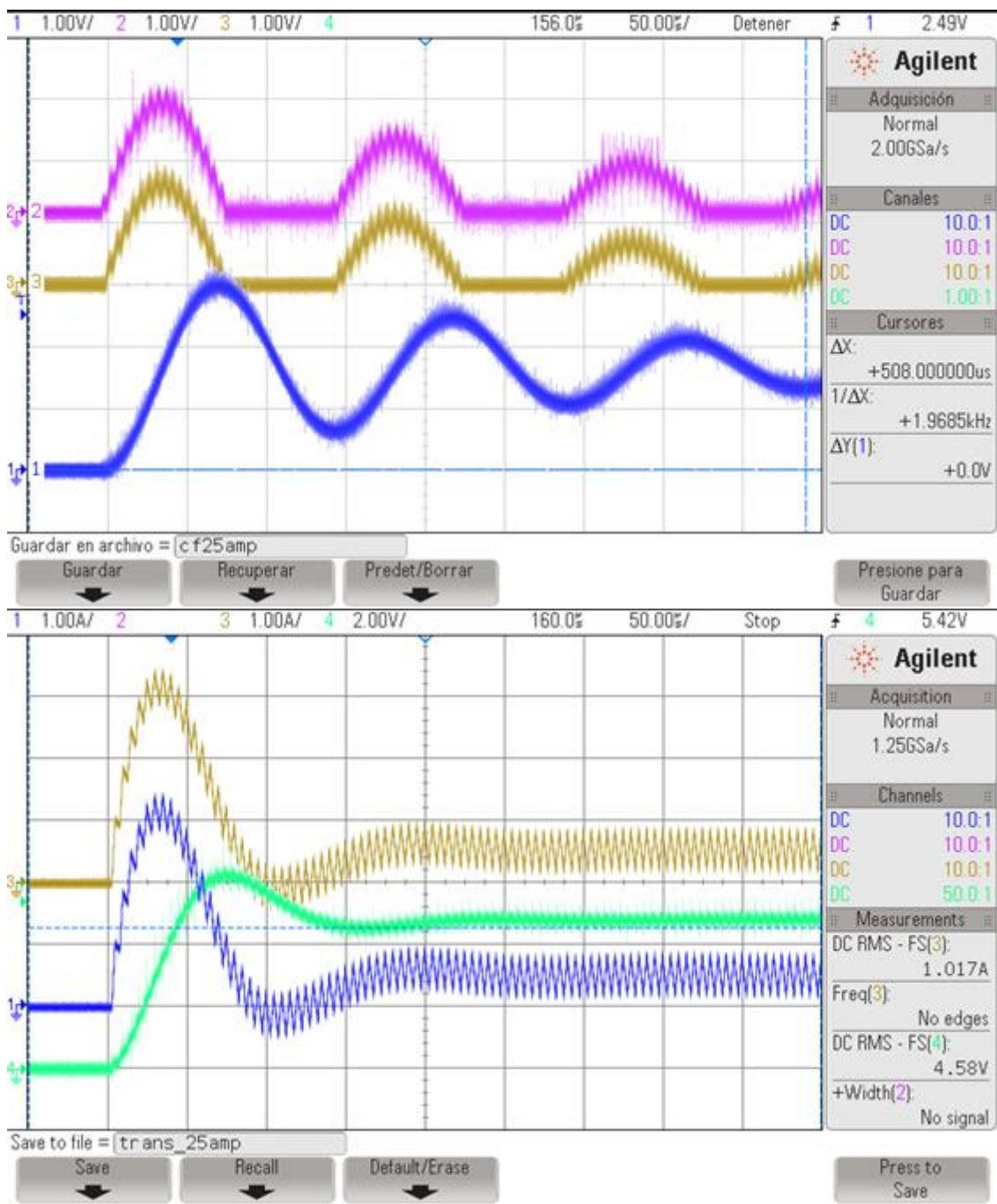


Figura 77: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 2,5 A

Corriente de salida de 5 A

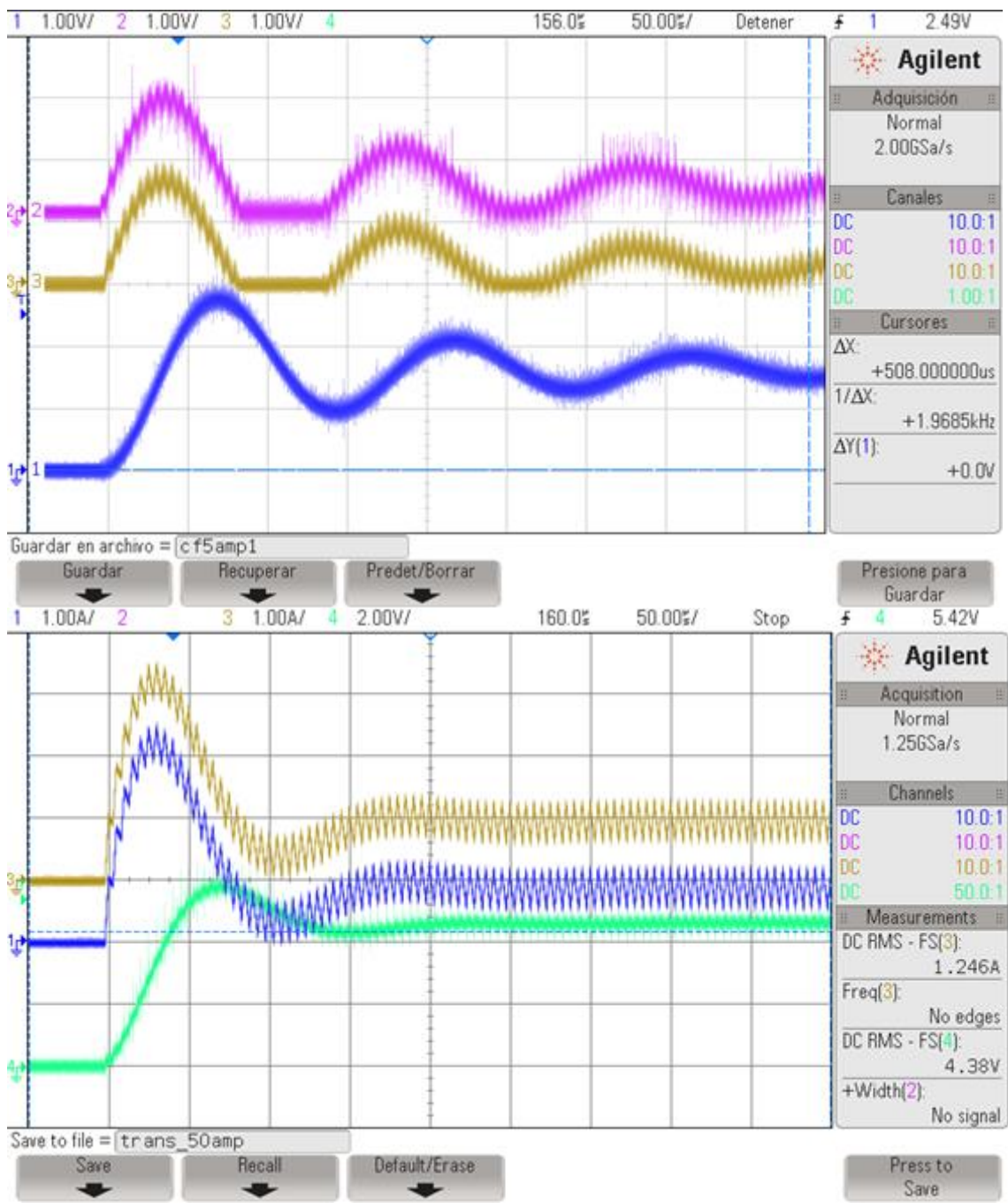


Figura 78: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 5 A

Corriente de salida de 8 A

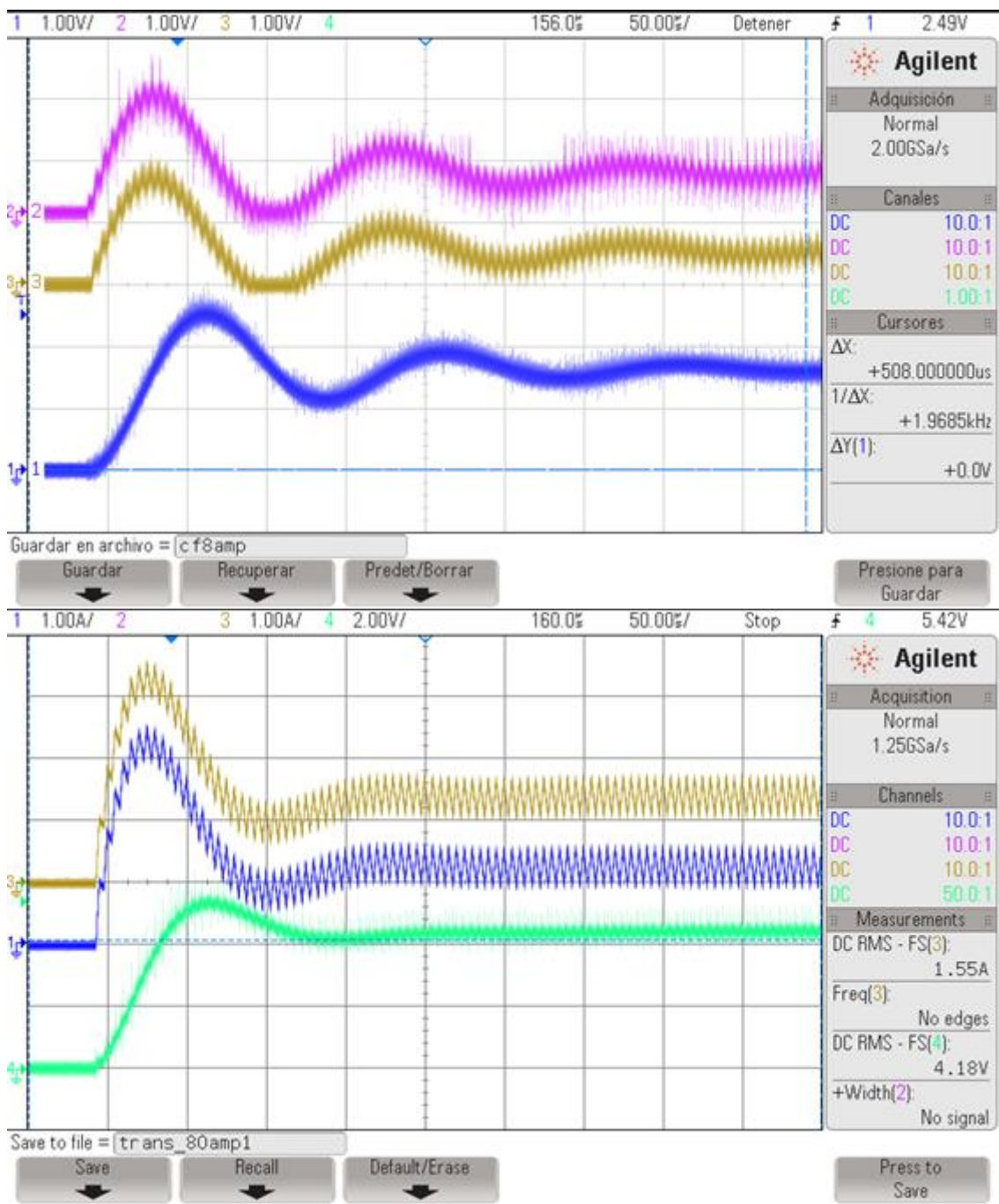


Figura 79: Comparativa de transitorios modelo HIL - realidad para corriente de salida de 8 A

ANEXO H Coste del módulo de conversión completo

Elemento	Cantidad	Precio unitario	Precio
Convertor digital-analógico DAC904U	1	17,1100 €	17,1100 €
Amplificador operacional CLC1005	1	1,0340 €	1,0340 €
Condensador 1 uF	2	0,4220 €	0,8440 €
Condensador 100 nF	6	0,0170 €	0,1020 €
Condensador 10 pF	2	0,0310 €	0,0620 €
Resistencia 2 kΩ	1	0,1220 €	0,1220 €
Resistencia 100 Ω	2	0,0118 €	0,0236 €
Resistencia 680 Ω	1	0,0118 €	0,0118 €
Resistencia 2k7 Ω	1	0,0118 €	0,0118 €
Resistencia 4k7 Ω	1	0,0118 €	0,0118 €
Resistencia 27 kΩ	1	0,0118 €	0,0118 €
Pines hembra individuales	4	-	0,1436 €
Pines macho individuales	11	-	1,0880 €
Conector IDC 24 pines hembra	2	1,2200 €	2,4400 €
Conector IDC 24 pines macho	1	4,7000 €	4,7000 €
PCB de dos capas	1	28,0900 €	28,0900 €
TOTAL:			55,8064 €

Tabla 9: Coste del módulo de conversión